

# **ANALISIS SENTIMEN KONTEN RADIKAL DI MEDIA SOSIAL TWITTER MENGGUNAKAN METODE *SUPPORT VECTOR MACHINE (SVM)***

## **SKRIPSI**

Untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer

Disusun oleh:  
Ferdie Alvianda  
NIM. 1150608000111006



PROGRAM STUDI TEKNIK INFORMATIKA  
JURUSAN TEKNIK INFORMATIKA  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS BRAWIJAYA  
MALANG  
2018

## PENGESAHAN

ANALISIS SENTIMEN KONTEN RADIKAL DI MEDIA SOSIAL TWITTER  
MENGUNAKAN METODE *SUPPORT VECTOR MACHINE* (SVM)

### SKRIPSI

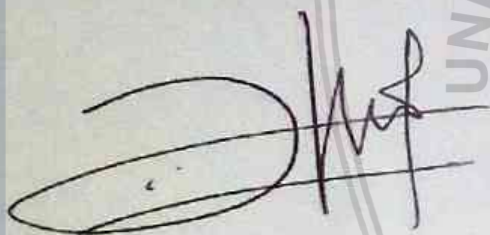
Untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer

Disusun Oleh :  
Ferd Alvianda  
NIM: 115060800111006

Skripsi ini telah diuji dan dinyatakan lulus pada  
03 Agustus 2018

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I



Indriati, S.T., M.Kom  
NIP: 19831013 201504 2 002

Dosen Pembimbing II



Putra Pandu Adikara, S.Kom, M.Kom  
NIP: 19850725 200812 1 002

Mengetahui

Ketua Jurusan Teknik Informatika



Tri Astoto Kurniawan, S.T., M.T., Ph.D.  
NIP. 19710518 200312 1 001

## IDENTITAS PENGUJI

Penguji 1 : Ir. Sutrisno, M.T  
NIP : 195703251987011001  
Penguji 2 : Faizatul Amalia, S.Pd.,M.Pd  
NIP : 201309 860821 2 001



## PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 3 Agustus 2018



Ferdi Alvianda

NIM : 115060800111006

## DAFTAR RIWAYAT HIDUP

TK Al Kautsar Pasuruan	: 1997-1999
SDN Kandang Sapi II Pasuruan	: 1999-2005
SMP Negeri 1 Pasuruan	: 2005-2008
SMA Negeri 1 Pasuruan	: 2008-2011





## UCAPAN TERIMA KASIH

Puji syukur saya ucapkan kepada Tuhan Yang Maha Esa, karena atas berkat rahmat-Nya penulis dapat menyelesaikan skripsi ini. Skripsi dengan judul “Analisis Sentimen Konten Radikal Di Media Sosial Twitter Menggunakan Metode *Support Vector Machine (SVM)*” disusun sebagai salah satu persyaratan untuk menyelesaikan studi pada jurusan Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya. Penulis menyadari bahwa skripsi ini dapat terselesaikan atas bantuan, dukungan dan bimbingan dari berbagai pihak yang telah membantu proses penyelesaiannya.



## ABSTRAK

**Ferdi Alvianda, Analisis Sentimen Konten Radikal Di Media Sosial Twitter Menggunakan Metode Support Vector Machine (SVM).**

**Pembimbing: Indriati, S.T, M.Kom, dan Putra Pandu Adikara, S.Kom, M.Kom.**

Di Indonesia akhir-akhir ini terjadi teror bom yang dilakukan orang-orang berpaham radikal. Orang-orang berpaham radikal kian hari kian bertambah jumlahnya dikarenakan orang-orang tersebut menyebarkan paham radikal kepada orang lain. Penyebaran konten radikal bisa dilakukan melalui media sosial, salah satunya adalah Twitter. Oleh sebab itu dilakukan penelitian terhadap konten radikal di media sosial Twitter. Dokumen Twitter yang berupa *tweet* radikal diklasifikasikan menjadi dua kategori, yaitu konten radikal positif dan konten radikal negatif. Metode yang digunakan pada penelitian ini adalah *Support Vector Machine* (SVM) dengan kernel *Polynomial Degree 2*. Tingkat akurasi tertinggi yang dihasilkan dari penelitian ini adalah 70% dengan nilai parameter  $\lambda$  sebesar 0,1, nilai konstanta  $\gamma$  sebesar 0,1, iterasi maksimum 5 dengan data latih sebanyak 80 dokumen (60 dokumen negatif dan 20 dokumen positif) untuk data latih dan 20 dokumen (15 dokumen negatif dan 5 dokumen positif) untuk data uji.

Kata Kunci : Analisis Sentimen, Twitter, *Support Vector Machine* (SVM).

## ABSTRACT

**Ferdi Alvianda, *Sentiment Analysis Radical Content Through Twitter Social Media Using Support Vector Machine (SVM)***

**Advisor: Indriati, S.T, M.Kom, and Putra Pandu Adikara, S.Kom, M.Kom.**

*Lately, there are many terrorist threats by the radicals in Indonesia. Radicals keep growing by numbers each day as they share their radical beliefs to other people. These radical beliefs can be shared through social media, such as Twitter. Therefore, a research regarding that problem is conducted. Documents of Twitter that contain radical tweets are classified to two categories, positive radical content and negative radical content. The method used for this research is Support Vector Machine (SVM) with Polynomial Degree Kernel. The highest accuracy rate achieved from this research is 70% with the parameter value of  $\lambda$  is 0,1, constant value of  $\gamma$  is 0,1, maximum iteration of 5 with training data sets of 80 documents (60 negative documents and 20 positive documents) as training data sets and 20 documents (15 negative documents and 5 positive documents) as testing data sets.*

**Keywords:** *Sentiment Analysis, Twitter, Support Vector Machine (SVM).*





## KATA PENGANTAR

Puji syukur kehadiran Allah SWT karena atas karunia serta hidayah-Nya lah penulis dapat menyelesaikan skripsi yang berjudul “Analisis Sentimen Konten Radikal pada Sosial Media Twitter Berbahasa Indonesia Menggunakan Metode *Support Vector Machine (SVM)*”. Shalawat dan salam semoga akan selalu tercurah kepada Nabi kita Muhammad Rasulullah SAW, keluarga, para sahabat, hingga umatnya sampai akhir zaman, amin.

Penulis menyadari bahwa penyusunan skripsi ini tidak akan terwujud tanpa adanya dukungan, bantuan, motivasi, serta doa dari berbagai pihak. Oleh karena itu, pada kesempatan ini penulis menyampaikan ucapan terima kasih kepada:

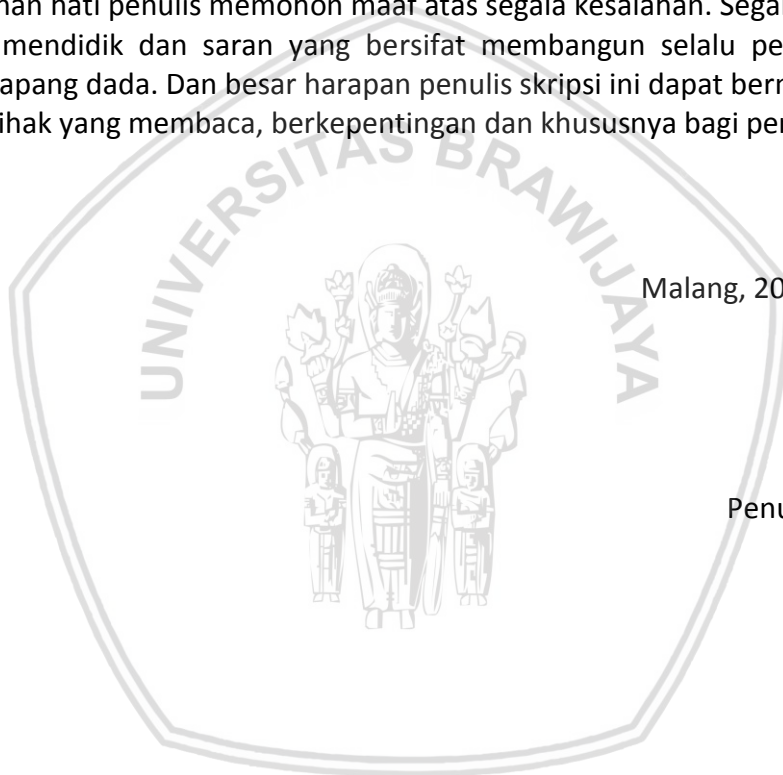
1. Ibu Indriati, S.T, M.Kom selaku Dosen Pembimbing I dan Bapak Putra Pandu Adikara, S.Kom, M.Kom selaku Dosen Pembimbing II yang bersedia meluangkan waktu untuk membimbing, membagi pengetahuan, kebijaksanaan dan motivasi, serta kritik dan saran dari awal hingga selesainya skripsi ini.
2. Bapak Imam Cholissodin, S.Si, M.Kom yang memberikan judul untuk penelitian ini.
3. Bapak Randy Cahya Wihandika, S.ST., M.Kom yang pernah menjadi Dosen Pembimbing II sebelumnya.
4. Bapak Wayan Firdaus Mahmudy, S.Si., M.T., Ph.D., Bapak Ir. Heru Nurwarsito, M.Kom., Bapak Uprapto S.T., M.T., Bapak Edy Santoso, S.Si., M.Kom. selaku Dekan, Wakil Dekan I, Wakil Dekan II dan Wakil Dekan III Fakultas Ilmu Komputer Universitas Brawijaya.
5. Seluruh dosen Fakultas Ilmu Komputer yang telah mendidik dan memberikan ilmu serta wawasannya selama menempuh pendidikan hingga menyelesaikan skripsi ini.
6. Dany Ardhian, S.Pd., M.Hum., dosen Sastra Indonesia selaku pakar dari Fakultas Ilmu Budaya Universitas Brawijaya yang bersedia meluangkan waktu dan memberikan bimbingan kepada penulis.
7. Seluruh civitas akademika Fakultas Ilmu Komputer Universitas Brawijaya yang telah banyak memberikan bantuan serta dukungan kepada penulis selama menempuh pendidikan hingga menyelesaikan skripsi ini.
8. Orang tua penulis yang senantiasa mendidik, memberikan motivasi, perhatian, dukungan serta doa yang tulus dan tidak pernah terputus kepada penulis.
9. Brian Andrianto dan Aulia Aida Rahmi yang sama-sama berjuang menyelesaikan penelitian ini, mencari data dan menemui pakar selama pengerjaan penelitian bersama penulis.

10. Seluruh teman-teman Informatika UB angkatan 2011 khususnya Reza, Indra, Maria, Etanto, Aula Rieza, Hangga, Monel, serta dan teman-teman di luar jurusan Informatika yang selalu menyemangati dan memberikan masukan terhadap penulis.
11. Seluruh teman-teman di luar Informatika UB khususnya Tito, Dimas, Deddy, Hasti, Anif, Sasing UM 2011, C2, CM6, PJ3, PRFC, Angels, GP Family, yang menyemangati dan membantu doa.
12. Semua pihak yang telah membantu dan tidak dapat penulis sebutkan satu per satu.

Meskipun telah berusaha dengan maksimal, penulis menyadari bahwa penulisan skripsi ini masih jauh dari sempurna. Oleh karena itu, dengan segala kerendahan hati penulis memohon maaf atas segala kesalahan. Segala kritik yang bersifat mendidik dan saran yang bersifat membangun selalu penulis terima dengan lapang dada. Dan besar harapan penulis skripsi ini dapat bermanfaat bagi semua pihak yang membaca, berkepentingan dan khususnya bagi penulis sendiri.

Malang, 20 Juli 2018

Penulis



## DAFTAR ISI

PENGESAHAN .....	i
IDENTITAS PENGUJI .....	ii
PERNYATAAN ORISINALITAS .....	iii
DAFTAR RIWAYAT HIDUP .....	iv
UCAPAN TERIMA KASIH .....	v
Abstrak .....	vi
Abstract .....	vii
KATA PENGANTAR .....	viii
DAFTAR ISI .....	x
DAFTAR TABEL .....	xiii
DAFTAR GAMBAR .....	xiv
DAFTAR KODE PROGRAM .....	xv
BAB 1 PENDAHULUAN .....	1
1.1 Latar belakang .....	1
1.2 Rumusan masalah .....	2
1.3 Tujuan .....	3
1.4 Manfaat .....	3
1.5 Batasan masalah .....	3
1.6 Sistematika Pembahasan .....	3
BAB 2 LANDASAN KEPUSTAKAAN .....	5
2.1 Kajian Pustaka .....	5
2.2 Dasar Teori .....	8
2.2.1 Twitter .....	8
2.2.2 Konten Radikal .....	8
2.2.3 <i>Text Preprocessing</i> .....	10
2.2.3.1 Tokenisasi .....	10
2.2.3.2 Filtering .....	10
2.2.4 Analisis Sentimen .....	11
2.2.5 <i>Support Vector Machine (SVM)</i> .....	11
2.2.6 <i>Sequential Training SVM</i> .....	13
2.2.7 Metode Evaluasi .....	14

BAB 3 METODOLOGI PENELITIAN .....	15
3.1 Tipe Penelitian .....	15
3.2 Strategi dan Rancangan Penelitian .....	15
3.2.1 Strategi Penelitian .....	15
3.2.2 Objek atau Partisipan Penelitian .....	15
3.2.3 Lokasi Penelitian .....	16
3.2.4 Metode/Teknik Pengumpulan Data .....	16
3.2.5 Metode/Teknik Analisis Data .....	16
3.2.6 Peralatan Pendukung .....	16
3.2.7 Metode/Teknik .....	17
BAB 4 PERANCANGAN DAN IMPLEMENTASI .....	18
4.1 Identifikasi Permasalahan .....	18
4.2 Diagram Alir Sistem .....	18
4.2.1 <i>Preprocessing</i> .....	19
4.2.1.1 Penandaan Frase .....	20
4.2.1.2 Tokenisasi .....	21
4.2.1.3 Filtering .....	22
4.2.2 Klasifikasi SVM .....	22
4.3 Perhitungan Manual .....	22
4.3.1 Data Set .....	23
4.3.2 <i>Preprocessing</i> .....	23
4.3.3 Metode SVM .....	24
4.4 Perancangan Antarmuka .....	31
4.5 Implementasi Sistem dan Antarmuka .....	32
4.5.1 Implementasi Sistem .....	32
4.5.1.1 Penandaan Frase .....	32
4.5.1.2 Tokenisasi .....	33
4.5.1.3 Filtering .....	33
4.5.1.4 Term Frequency .....	34
4.5.1.5 Term Frequency (Normalisasi) .....	35
4.5.1.6 Proses Perhitungan Kernel .....	36
4.5.1.7 Proses Perhitungan Matriks Hessian .....	37
4.5.1.8 Proses Iterasi .....	38

4.5.1.9 Proses Perhitungan E .....	38
4.5.1.10 Proses Perhitungan Delta Alfa .....	39
4.5.1.11 Proses Perhitungan Alfa Baru .....	41
4.5.1.12 Proses Perhitungan Support Vector.....	42
4.5.1.13 Proses Perhitungan Bias(b) .....	44
4.5.1.14 Proses Testing .....	44
4.5.2 Implementasi Antarmuka .....	45
4.5.2.1 Antarmuka Halaman Preprocessing.....	46
4.5.2.2 Implementasi Metode SVM .....	47
4.5.2.3 Antarmuka Halaman Pengujian .....	49
4.6 Skenario Pengujian .....	49
BAB 5 Pengujian dan analisis .....	51
5.1 Pengujian .....	51
5.2 Hasil Pengujian.....	51
5.2.1 Hasil Pengujian Terhadap Nilai Parameter $\lambda$ .....	51
5.2.2 Hasil Pengujian Terhadap Nilai Konstanta $\gamma$ .....	52
5.2.3 Hasil Pengujian Terhadap Iterasi Maksimum.....	53
5.3 Analisis Pengujian .....	54
BAB 6 Penutup .....	57
6.1 Kesimpulan.....	57
6.2 Saran .....	57
DAFTAR PUSTAKA.....	58
LAMPIRAN A DATA YANG DIGUNAKAN UNTUK SISTEM.....	60
A.1 Data Latih.....	60
A2. Data Uji .....	64
A3. Daftar Kata Yang Ditandai .....	66
A4. Daftar Kata Yang Dihapus .....	93

## DAFTAR TABEL

Tabel 2.1 Tabel Kajian Pustaka.....	5
Tabel 4.1 Tabel Data Set Dokumen Manualisasi.....	23
Tabel 4.2 Data Set <i>Feature Term</i> .....	23
Tabel 4.3 <i>Preprocessing</i> .....	24
Tabel 4.4 Tabel Perhitungan <i>Term Frequency</i> .....	24
Tabel 4.5 Tabel Normalisasi <i>Term Frequency</i> .....	25
Tabel 4.6 Parameter yang Digunakan .....	25
Tabel 4.7 Hasil Perhitungan Kernel.....	26
Tabel 4.8 Hasil Perhitungan Matriks <i>Hessian</i> .....	26
Tabel 4.9 Hasil Perhitungan E .....	27
Tabel 4. 10 Hasil Herhitungan $\delta\alpha_i$ .....	27
Tabel 4.11 Hasil Perhitungan $\alpha_i$ .....	28
Tabel 4.12 Hasil Perhitungan $\alpha_i$ .....	28
Tabel 4.13 Pemilihan $\alpha_i$ Terbesar.....	28
Tabel 4.14 Perhitungan Kernel dari $\alpha_i$ Terbesar .....	29
Tabel 4.15 Hasil Perhitungan $W.X$ .....	29
Tabel 4.16 Tabel <i>Testing</i> Data 1 .....	30
Tabel 4.17 Tabel <i>Testing</i> Data 2 .....	30
Tabel 4.18 Tabel <i>Testing</i> Data 3 .....	30
Tabel 4.19 Tabel <i>Testing</i> Data 4 .....	31
Tabel 4.20 Tabel Pengujian 1 .....	49
Tabel 4.21 Tabel Pengujian 2 .....	50
Tabel 4.22 Tabel Pengujian 3 .....	50
Tabel 5.1 Hasil Pengujian Terhadap Nilai Parameter $\lambda$ .....	51
Tabel 5.2 Dokumen Relevan Pada Pengujian Terhadap Nilai Parameter $\lambda$ .....	52
Tabel 5.3 Hasil Pengujian Terhadap Nilai Konstanta $\gamma$ .....	52
Tabel 5.4 Dokumen Relevan Pada Pengujian Terhadap Nilai Konstanta $\gamma$ .....	53
Tabel 5.5 Hasil Pengujian Terhadap Iterasi Maksimum.....	54
Tabel 5.6 Dokumen Relevan Pada Pengujian Terhadap Iterasi Maksimum .....	54



## DAFTAR GAMBAR

Gambar 2.1 Contoh <i>Tweet</i> .....	8
Gambar 2.2 Ilustrasi Tokenisasi .....	10
Gambar 2.3 Ilustrasi <i>Filtering</i> .....	11
Gambar 2.4 Ilustrasi SVM.....	12
Gambar 4.1 Diagram Alir Sistem Secara Umum .....	18
Gambar 4.2 Diagram Alir Sistem <i>Preprocessing</i> .....	19
Gambar 4.3 Diagram Alir Sistem Menandai Frase .....	20
Gambar 4.4 Diagram Alir Sistem Proses <i>Tokenizing</i> .....	21
Gambar 4.5 Diagram Alir Sistem Proses <i>Filtering</i> .....	22
Gambar 4.6 Perancangan Antarmuka.....	31
Gambar 4.7 Antarmuka Halaman <i>Preprocessing</i> (1).....	46
Gambar 4.8 Antarmuka Preprocecing (2) .....	47
Gambar 4.9 Antarmuka Pelatihan (1) .....	47
Gambar 4.10 Antarmuka Halaman Pelatihan (2).....	48
Gambar 4.11 Antarmuka Halaman Pelatihan (3).....	48
Gambar 4.12 Antarmuka Halaman Pengujian .....	49
Gambar 5.1 Pengujian Nilai Parameter $\lambda$ .....	55
Gambar 5.2 Pengujian Nilai Konstanta $\gamma$ .....	56
Gambar 5.3 Pengujian Iterasi Maksimum.....	56

## DAFTAR KODE PROGRAM

Kode Program 4.1 Penandaan Frase .....	32
Kode Program 4.2 Tokenisasi .....	33
Kode Program 4.3 <i>Filtering</i> .....	33
Kode Program 4.4 <i>Term Frequency</i> .....	34
Kode Program 4.5 <i>Term Frequency</i> (Normalisasi) .....	36
Kode Program 4.6 Perhitungan Kernel .....	36
Kode Program 4.7 Perhitungan Matriks Hessien .....	38
Kode Program 4.8 Iterasi .....	38
Kode Program 4.9 Perhitungan E .....	39
Kode Program 4.10 Perhitungan Delta Alfa .....	41
Kode Program 4.11 Perhitungan Alfa Baru .....	42
Kode Program 4.12 Perhitungan <i>Support Vector</i> .....	43
Kode Program 4.13 Perhitungan Bias .....	44
Kode Program 4.14 <i>Testing</i> .....	45

## BAB 1 PENDAHULUAN

Bab ini membahas tentang latar belakang, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan masalah, dan sistematika penulisan skripsi.

### 1.1 Latar belakang

Manusia sejatinya diciptakan sebagai makhluk sosial yang butuh akan sosialisasi dan komunikasi antar sesamanya (Anggraeni, 2013). Dengan perkembangan internet, semakin banyak hal yang ditawarkan di dalamnya. Mulai dari informasi terkini, data dan statistik, pengetahuan, dan media sosial. Saat ini media sosial diposisikan sebagai ruang publik baru. Dengan semakin terbatasnya waktu untuk bersosialisasi, media sosial dianggap mampu memuaskan manusia untuk menyampaikan apa yang ingin mereka sampaikan (Anggraeni, 2013). Adapun media sosial yang paling banyak digunakan saat ini adalah Facebook, Twitter, dan Instagram. Khusus untuk Twitter, Indonesia menjadi negara terbesar kelima dalam hal banyaknya pengguna Twitter (Asih, 2012).

Twitter adalah penyedia layanan pesan yang menyediakan begitu banyak karakteristik dengan alat komunikasi yang digunakan (O'Reilly & Milstein, 2011). Twitter termasuk salah satu media sosial yang termasuk simpel. Melalui Twitter seseorang bisa menyampaikan apa yang ingin disampaikan. Mulai dari cerita tentang apa yang dialaminya hari ini, keluhan, dan opini, semuanya bisa dilakukan dengan mudah (Anggraeni, 2013). Apalagi dengan banyaknya aplikasi *web* dan perangkat *mobile* yang beredar, Twitter tidak hanya digunakan sebagai media untuk menyampaikan apa yang seseorang ingin sampaikan saja. Twitter juga digunakan sebagai sarana bisnis, pencitraan, media massa, dan sebagainya.

Saat ini, Twitter bisa dijadikan sebuah indikator yang baik untuk memberikan pengaruh di dalam sebuah penelitian (Putranti, 2014). Isi *tweet* di Twitter dijadikan objek penelitian. Namun tidak semua *tweet* dijadikan objek penelitian. Biasanya *tweet* hanya topik-topik tertentu yang dijadikan objek penelitian, seperti *tweet* yang berhubungan dengan dunia politik, budaya, pemasaran, dan sebagainya. Tak sedikit perusahaan dan tokoh-tokoh berpengaruh yang mempunyai akun Twitter untuk menyapa dan berinteraksi dengan pengguna lain di media sosial Twitter. Mereka juga tak segan-segan meminta komentar atau *feedback* melalui akun Twitternya.

Beberapa waktu lalu, di Indonesia terjadi teror bom di daerah Sarinah, Jakarta. Tidak hanya kejadian itu saja, Indonesia sebelumnya menjadi korban juga pada tahun 2002 dan 2005 di Bali, dan 2004 di Kuningan Jakarta. Selain kejadian di Indonesia, dunia juga dihebohkan ketika ada aksi terorisme dari kaum radikal di Paris. Teror tersebut dilakukan oleh golongan ekstrimis/radikal. Para pelaku aksi radikal semakin hari terus bertambah jika dilihat jumlah kasus yang melibatkan para pelaku radikal. Dari satu kasus misalnya, bom bali, semua pelakunya sudah berhasil ditangkap atau mati saat dilakukan penangkapan. Tapi

kemudian muncul kasus lain yang melibatkan para pelaku yang radikal. Para pelaku radikal tersebut terus bertambah dari waktu ke waktu karena terjadi penyebaran paham radikal.

Penyebaran paham radikal sendiri bisa dilakukan dengan banyak cara, bisa melalui ceramah, diskusi ataupun melalui media sosial. Melalui media sosial, penyebaran paham radikal bisa berupa konten radikal. Semua konten entah berupa gambar, tulisan, audio, ataupun video yang mengandung paham radikal disebut konten radikal sehingga tidak menutup kemungkinan bisa dilakukan penelitian tentang analisis sentimen terhadap *tweet* berbahasa Indonesia tentang konten radikal.

Analisis sentimen yang juga disebut dengan *opinion mining* merupakan bidang studi yang menganalisis pendapat, penilaian, evaluasi, sentimen, sikap dan emosi terhadap entitas seperti produk, jasa, organisasi, individu, peristiwa, topik dan atribut lainnya. Sejak awal tahun 2000, analisis sentimen telah berkembang menjadi salah satu penelitian aktif dalam *natural language processing*. Dalam beberapa tahun terakhir, analisis sentimen pada kegiatan industri juga terus berkembang (Liu, 2012). Banyaknya penelitian mengenai analisis sentimen disebabkan karena analisis sentimen memberikan manfaat yang cukup besar. Contohnya adalah untuk membantu pemantauan terhadap suatu produk, selain itu juga dapat digunakan sebagai alat bantu untuk mengetahui respon masyarakat terhadap produk tersebut. Seperti penelitian tentang analisis sentimen yang dilakukan oleh Novantirani (2015) dengan objek penggunaan transportasi darat umum di Twitter. Penelitian tersebut bertujuan untuk membantu dalam penilaian dan evaluasi terhadap penggunaan transportasi umum darat dalam kota. Penelitian tersebut menggunakan metode SVM.

SVM adalah salah satu metode yang bisa digunakan untuk pengklasifikasian. Banyak penelitian sebelumnya yang menggunakan metode ini sebagai metode untuk mengklasifikasi. Pada penelitian yang dilakukan Novantirani (2015), akurasi yang dihasilkan dengan metode SVM sebesar 78,12%. Selain itu, ada juga penelitian yang dilakukan oleh Putranti (2014) dengan tema yang sama yaitu analisis sentimen terhadap *tweet* berbahasa Indonesia. Pada penelitian yang juga menggunakan algoritme SVM memiliki akurasi sebesar 86,81%.

Berdasarkan uraian di atas, penulis berinisiatif untuk merancang suatu penelitian yang diharapkan dapat memberikan solusi dari permasalahan yang telah diuraikan.

## 1.2 Rumusan masalah

Berdasarkan latar belakang yang sudah diuraikan, dapat dirumuskan permasalahan penelitian ini adalah sebagai berikut.

1. Bagaimana mengimplementasikan metode *Support Vector Machine* (SVM) untuk analisis sentimen konten radikal di media sosial Twitter?

2. Bagaimana akurasi pengujian terhadap analisis sentimen terhadap konten radikal di media sosial Twitter menggunakan metode *Support Vector Machine* (SVM)?

### 1.3 Tujuan

Tujuan yang ingin dicapai dalam penelitian ini antara lain:

1. Membuat aplikasi analisis sentimen konten radikal pada dokumen twitter berbahasa Indonesia menggunakan metode *Support Vector Machine* (SVM).
2. Mengetahui tingkat akurasi penggunaan metode *Support Vector Machine* (SVM) dalam analisis sentimen konten radikal.

### 1.4 Manfaat

Manfaat yang diberikan dari penelitian ini adalah sebagai berikut.

1. Peneliti dapat lebih memahami tentang teknologi informasi dalam bidang *information retrieval* dan analisis sentimen.
2. Peneliti memahami lebih dalam mengenai pengimplementasian metode *Support Vector Machine* (SVM).
3. Sebagai referensi mahasiswa lain untuk memahami analisis sentimen dan metode *Support Vector Machine* (SVM).

### 1.5 Batasan masalah

Untuk menghindari meluasnya pembahasan dalam penelitian ini, maka perlu adanya batasan-batasan untuk menyederhanakan permasalahan, yaitu:

1. Algoritme yang digunakan adalah algoritme *Support Vector Machine* (SVM) dengan kernel *polynomial degree 2* dan tidak membandingkan dengan algoritme lain.
2. Data yang digunakan adalah dokumen *tweets* pada yang bersumber dari media sosial Twitter berbahasa Indonesia.
3. Data yang digunakan telah ditentukan sebelumnya, berdasarkan hasil diskusi dengan ahli bahasa.
4. Data memiliki dua kelas yaitu kelas positif dan kelas negatif.

### 1.6 Sistematika Pembahasan

Penyusunan skripsi ini berdasarkan sistematika pembahasan sebagai berikut.

#### BAB I Pendahuluan

Bab ini diisi dengan latar belakang, rumusan masalah, tujuan, manfaat, batasan masalah, dan sistematika penulisan.

**BAB II Landasan Kepustakaan**

Pada bab ini diuraikan teori-teori yang digunakan serta kajian pustaka yang mengenai perancangan dan pembuatan sistem analisis sentimen konten radikal pada dokumen Twitter dengan metode SVM dan dasar teori yang digunakan.

**BAB III Metodologi**

Membahas tentang gambaran umum tentang jenis penelitian dan strategi dan rancangan penelitian.

**BAB IV Perancangan dan Implementasi**

Membahas tentang perancangan dan implemtasi dari perancangan yang dibuat untuk sistem analisis sentimen konten radikal pada dokumen Twitter dengan metode SVM berdasarkan metodologi yang telah dibuat.

**BAB V Pengujian dan Analisis**

Memuat pengujian, hasil pengujian, dan analisis terhadap sistem.

**BAB VI Penutup**

Memuat kesimpulan dari penelitian yang telah dilakukan serta saran untuk pengembangan lebih lanjut.



## BAB 2 LANDASAN KEPUSTAKAAN

Bab ini menjelaskan tentang kajian pustaka dan dasar teori yang digunakan untuk menunjang penyusunan skripsi. Kajian pustaka berisi tentang pembahasan penelitian sebelumnya yang berkaitan dengan skripsi ini. Sedangkan dasar teori berisi mengenai pembahasan teori yang digunakan untuk menyusun penelitian. Adapun dasar teori yang dibahas antara lain: Twitter, konten radikal, *text preprocessing*, analisis sentimen, algoritme SVM, dan evaluasi.

### 2.1 Kajian Pustaka

Algoritme SVM sebelumnya telah digunakan untuk banyak penelitian sebelumnya dengan studi kasus analisis sentimen pada topik yang berbeda. Pada sub bab ini, penulis akan membahas penelitian sebelumnya yang mendukung penelitian ini. Uraian tentang penelitian sebelumnya dan yang diusulkan dapat dilihat pada Tabel 2.1.

**Tabel 2.1 Tabel Kajian Pustaka**

No.	Penelitian	Objek	Metode	Hasil
1.	Novantirani, A., 2015	<i>Tweet</i> Bahasa Indonesia tentang menggunakan angkutan umum	<i>Support Vector Machine</i> (SVM)	Penelitian ini berfokus pada analisis sentimen tentang penggunaan transportasi umum darat pada media sosial Twitter menggunakan metode SVM. Penelitian ini menunjukkan bahwa metode SVM dapat digunakan untuk mengklasifikasi sentimen/opini pada media sosial Twitter yang apabila dilakukan dengan metode konvensional membutuhkan waktu yang lama. Akurasi dari pengimplemtasian metode SVM pada penelitian ini sebesar 78,12%.
2.	Putranti, N. D., 2014	<i>Tweet</i> Bahasa Indonesia yang disertai <i>emoticon</i>	<i>Support Vector Machine</i> (SVM)	Penelitian ini berfokus pada analisis sentimen pada media sosial Twitter untuk mengklasifikasi <i>tweet</i> menggunakan metode SVM. Sebelum dilakukan klasifikasi, terlebih dahulu akan dilakukan <i>POS tagger</i> dengan menggunakan <i>Maximum entropy</i> . Penelitian ini menunjukkan bahwa melakukan <i>POS tagger</i> dan metode SVM dapat digunakan untuk mengklasifikasi dokumen Twitter dengan bukti akurasi yang cukup tinggi, yaitu 86,81%.

No.	Penelitian	Objek	Metode	Hasil
3.	Khan, A. Z. H., 2015.	Komentar dan <i>tweet</i>	<i>Support Vector Machine</i> (SVM)	Penelitian ini berfokus pada analisis sentimen pada komentar dan media sosial Twitter untuk mengklasifikasi komentar atau <i>tweet</i> positif dan negatif menggunakan metode SVM. Penelitian ini menunjukkan bahwa metode SVM dapat digunakan untuk mengklasifikasi komentar atau <i>tweet</i> yang positif dan negatif yang apabila dilakukan dengan metode konvensional membutuhkan waktu yang lama.
4.	Patil, G. et al., 2014	Komentar dan <i>tweet</i> tentang kandidat politik	<i>Support Vector Machine</i> (SVM) dan metode ANN	Penelitian ini berfokus pada analisis sentimen pada komentar dan media sosial Twitter untuk mengklasifikasi komentar atau <i>tweet</i> terhadap kandidat politik menggunakan metode SVM. Penelitian ini menunjukkan bahwa metode SVM memberikan hasil yang bagus dalam mengklasifikasi komentar atau <i>tweet</i> jika dibandingkan dengan metode ANN.
5.	Analisis Sentimen Konten Radikal di Media Sosial Twitter Menggunakan Metode SVM	<i>Tweet</i> Bahasa Indonesia tentang konten radikal	<i>Support Vector Machine</i> (SVM)	Penelitian ini diusulkan untuk mengklasifikasi sentimen/opini masyarakat pada media sosial Twitter tentang konten radikal untuk memudahkan dalam mengklasifikasi konten radikal positif dan negatif. Metode SVM yang digunakan adalah SVM kernel <i>Polynomial Degree 2</i> .

Penelitian pertama adalah penelitian yang dilakukan oleh Novantirani pada tahun 2015. Penelitian ini berfokus pada opini di Twitter tentang penggunaan transportasi darat umum. Tujuan dari penelitian tersebut adalah untuk membantu dalam penilaian dan evaluasi terhadap penggunaan transportasi umum darat dalam kota. Sistem yang dirancang pada penelitian ini menerima inputan berupa *dataset* Twitter bertipe file *Excel*. Data tersebut diambil dari berdasarkan input *query* menggunakan *tools NodeXL*. *Dataset* ini dikelompokkan menjadi empat sesuai jenis transportasi umum darat dalam kota. Setelah itu, data tadi dilakukan *preprocessing* dengan tujuan membuat bentuk *dataset* sesuai dengan kebutuhan klasifikasi dan memudahkan pemrosesan pada sistem. Untuk pencarian hasil klasifikasi sentimen dilakukan proses *stopword removal*, yaitu untuk menghapus kata yang sering muncul tetapi tidak memiliki makna spesifik. Kemudian dilakukan pembobotan kata berdasarkan jumlah kemunculan dalam dokumen, sehingga dokumen dapat direpresentasikan dalam vektor. Fitur

pembobotan yang digunakan yaitu *unigram*, dan metode pembobotan *Term Frequency-Inverse Document Frequency* (TF-IDF). Proses selanjutnya adalah pengklasifikasian sentimen data dengan menggunakan metode *Support Vector Machine* (SVM). Akurasi penelitian ini adalah 78,12% (Novantirani, 2015).

Penelitian kedua adalah penelitian yang dilakukan oleh Putranti pada tahun 2014. Data yang digunakan berjumlah 81.885 *tweet*. Setelah melalui proses *preprocessing* dan *POS tagging* data tadi menjadi berjumlah 44.006 *tweet*. Dalam melakukan *preprocessing* ada empat tahapan yang dilakukan, yaitu *cleaning* (proses untuk membersihkan dokumen dari kata-kata yang tidak diperlukan), *case folding* (penyeragaman bentuk huruf serta penghilangan tanda baca), *parsing* (memecah dokumen menjadi kata per kata dengan menganalisis terhadap kumpulan kata dengan memisahkan kata tersebut dan menentukan struktur sintaksis dari tiap kata tersebut), dan filter bahasa (proses pemilihan *tweet* yang berbahasa Indonesia saja, jika ditemukan kata berbahasa Indonesia tidak baku maka kata tersebut diganti dengan sinonimnya berupa kata baku yang sesuai dengan Kamus Besar Bahasa Indonesia). Selanjutnya akan dilakukan POS tagger dengan menggunakan *Maximum entropy*. Setelah itu, barulah dilakukan klasifikasi dengan metode SVM. Untuk mendapatkan hasil terbaik, diujikan menggunakan menggunakan fungsi linier dan tiga kernel berbeda, yaitu *Polynomial*, *RBF* dan *Sigmoid*. Untuk pengujian *7-fold cross validation* dengan tipe kernel *Sigmoid* diperoleh akurasi sebesar 86,81% yang merupakan hasil uji terbaik (Putranti, 2014).

Penelitian ketiga adalah penelitian yang dilakukan Khan pada tahun 2015. Penelitian ini dilakukan untuk membedakan opini dari dokumen atau *tweet* positif dan negatif. Tahapan pertama yang dilakukan adalah *preprocessing*. *Preprocessing* yang dilakukan antara lain tokenisasi, *stopword removal*, dan *stemming*. Setelah itu akan dilakukan pembobotan dengan TF-IDF. Lalu dilakukan *feature selection* agar klasifikasi menjadi lebih efektif. *Feature selection* dilakukan dengan cara menganalisa dan mengidentifikasi *feature* yang relevan. Setelah itu barulah dilakukan klasifikasi dengan metode SVM. Dari penelitian ini, dapat disimpulkan bahwa semakin banyak data yang digunakan, semakin tinggi akurasi dari metode SVM (Khan, 2015).

Penelitian keempat adalah penelitian yang dilakukan Patil, dkk pada tahun 2014. Penelitian ini dilakukan untuk membedakan opini dari dokumen atau *tweet* positif dan negatif dengan objek kandidat politik. Sama seperti penelitian ketiga, tahapan pertama yang dilakukan adalah *preprocessing*. *Preprocessing* yang dilakukan antara lain tokenisasi, *stopword removal*, dan *stemming*. Setelah itu akan dilakukan pembobotan dengan TF-IDF. Lalu dilakukan *feature selection* agar klasifikasi menjadi lebih efektif. *Feature selection* dilakukan dengan cara menganalisis dan mengidentifikasi *feature* yang relevan. Setelah itu barulah dilakukan klasifikasi dengan metode SVM. Pada penelitian ini, peneliti menyimpulkan bahwa metode SVM memberikan hasil yang bagus dalam mengklasifikasi komentar atau *tweet* jika dibandingkan dengan metode ANN (Patil, et al., 2014).

## 2.2 Dasar Teori

Dasar teori yang digunakan dalam penelitian meliputi Twitter, konten radikal, *text preprocessing*, analisis sentimen, algoritme SVM, dan evaluasi.

### 2.2.1 Twitter

Twitter adalah penyedia layanan pesan yang menyediakan begitu banyak karakteristik dengan alat komunikasi yang digunakan (O'Reilly & Milstein, 2011). Twitter adalah salah satu media sosial yang *termasuk* simpel. Melalui Twitter seseorang bisa menyampaikan apa yang ingin dia sampaikan. Mulai dari cerita tentang apa yang dialaminya hari ini, keluhan, dan opini, semuanya bisa dilakukan dengan mudah (Anggraeni, 2013). Apalagi dengan banyaknya aplikasi *web* dan perangkat *mobile* yang beredar, Twitter tidak hanya digunakan sebagai media untuk menyampaikan apa yang seseorang ingin sampaikan saja. Twitter juga digunakan sebagai sarana bisnis, pencitraan, media massa, dan sebagainya. Gambar 2.1 merupakan contoh penggunaan Twitter.



**Gambar 2.1 Contoh Tweet**

Adapun istilah-istilah umum di dalam Twitter antara lain:

1. *Tweeps* adalah istilah pengguna Twitter,
2. *Tweet* adalah istilah untuk *post*/komentar/kicauan di Twitter. Pada Gambar 2.1 diilustrasikan contoh *tweet*,
3. *Retweet* adalah istilah untuk men-*tweet* ulang *tweet* pengguna lain untuk dimasukkan ke dalam *profile*-nya,
4. *Reply* adalah fitur yang digunakan untuk membalas *tweet* pengguna lain,
5. *Quote* adalah fitur untuk memberikan komentar *tweet* pengguna lain,
6. *Hashtag* adalah penyertaan tanda '#' pada *tweet* agar mudah dikategorikan dan memudahkan dalam pencarian.

### 2.2.2 Konten Radikal

Di masa penjajahan Belanda di Indonesia, kata radikal memiliki makna yang positif. Menurut Adnan Buyung Nasution di dalam disertasinya di Utrecht, Belanda, pada tahun 1918, di Indonesia dibentuk apa yang disebut sebagai "*Radicale Concentratie*". *Radicale Concentratie* adalah organisasi yang terdiri

atas Budi Utomo, Sarekat Islam, Insulinde, dan Indische Sociaal Democratische Vereniging. Tujuannya dari pembentukan tersebut adalah untuk membentuk parlemen yang terdiri atas wakil-wakil yang dipilih dari kalangan rakyat (Husaini, 2005).

Namun sekarang malah sebaliknya, radikal bermakna negatif. Di dalam Kamus Besar Bahasa Indonesia, aksi berarti gerakan atau tindakan. Sedangkan radikal berarti secara mendasar. Jika dihubungkan dengan politik, radikal berarti amat keras menuntut perubahan. Radikalisme berarti paham atau aliran yang menginginkan perubahan atau pembaharuan sosial dan politik dengan cara kekerasan atau drastis. Jadi, aksi radikal bisa artikan sebagai gerakan atau tindakan yang menginginkan perubahan dengan cara kekerasan. Salah satu contoh aksi radikal adalah tawuran, demo yang ricuh, terorisme.

Jika membahas radikalisme atau aksi radikal, kebanyakan orang akan menghubungkan istilah tersebut dengan agama. Padahal selain agama, radikalisme juga sudah “menjangkiti” aliran-aliran sosial, politik, budaya, dan ekonomi. Tapi bagi kalangan masyarakat awam radikalisme hanya dilakukan oleh agama tertentu saja. Dan anggapan itu memang tidak salah karena kelompok radikal di negeri ini tumbuh subur dan masih bebas melancarkan serangan.

Contoh aksi radikal yang berhubungan dengan agama terjadi pada akhir-akhir ini, yaitu aksi bom bunuh diri. Contoh aksi radikal dalam hal sosial adalah tawuran antar pelajar, tawuran antar geng. Contoh aksi radikal dalam hal politik adalah saling serangnya kaum elite politik dengan lawan politiknya. Contoh aksi radikal dalam hal ekonomi adalah adanya demo buruh yang menginginkan kenaikan gaji.

Para pelaku aksi radikal sendiri semakin hari terus bertambah jika dilihat jumlah kasus yang melibatkan para pelaku radikal. Dari satu kasus misalnya, bom bali, semua pelakunya sudah berhasil ditangkap atau mati saat dilakukan penangkapan. Kemudian muncul kasus lain yang melibatkan para pelaku yang radikal. Para pelaku radikal tersebut terus bertambah dari waktu ke waktu karena terjadi penyebaran paham radikal. Paham radikal sendiri penyebarannya bisa melalui media sosial. Semua konten entah berupa gambar, tulisan, audio, ataupun video yang mengandung paham radikal disebut konten radikal.

Menurut Badan Nasional Penanggulangan Terorisme (BNPT), suatu konten dianggap radikal ketika konten tersebut mengandung satu atau lebih kriteria di bawah ini: (Iradat, 2015)

1. Mengajak atau mengarahkan tindakan anarkis dan terorisme,
2. Mengandung unsur SARA (suku, agama dan ras),
3. Takfiri (mengkafirkan seseorang atau kelompok),
4. Memaknai jihad secara terbatas.

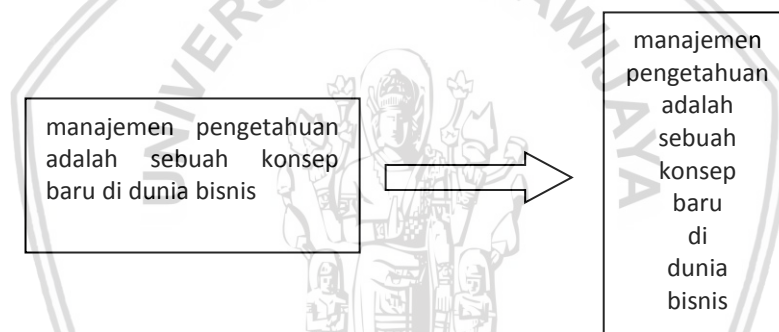


### 2.2.3 Text Preprocessing

Sebelum melakukan klasifikasi teks di dalam dokumen, ada tahapan yang harus dilakukan, yaitu *text preprocessing*. *Text preprocessing* sendiri ada beberapa tahapan, yaitu tokenisasi, *stemming*, dan *filtering*.

#### 2.2.3.1 Tokenisasi

Pada dasarnya tokenisasi adalah menghilangkan tanda baca dan memisahkan antara kata satu dengan kata lainnya. Saat melakukan tokenisasi dilakukan proses pemisahan suatu rangkaian karakter berdasarkan karakter spasi, dan mungkin pada waktu yang bersamaan dilakukan juga proses penghapusan karakter tertentu, seperti tanda baca (Amin, 2013). Token seringkali disebut sebagai istilah (*term*) atau kata, sebagai contoh sebuah token merupakan suatu urutan karakter dari dokumen tertentu yang dikelompokkan dan berguna untuk diproses. Seperti contoh, kata-kata “*computer*”, “*computing*”, dan “*compute*” semua berasal dari *term* yang sama yaitu “*comput*”, tanpa pengetahuan sebelumnya dari morfologi bahasa Inggris. Pada Gambar 2.2 diilustrasikan proses tokenisasi.



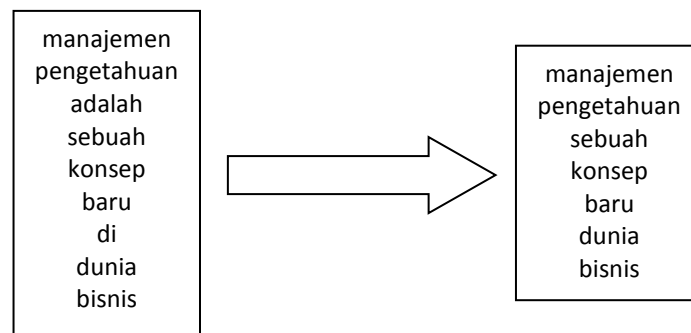
Gambar 2.2 Ilustrasi Tokenisasi

Sumber: (Merdekawati, 2014)

#### 2.2.3.2 Filtering

Proses *filtering* adalah proses pengeliminasian kata-kata yang dianggap tidak penting. Metode ini memiliki banyak keuntungan, yaitu akan mengurangi *space* pada *tabel term index* hingga 40% atau lebih. Tahap *filtering* dapat menggunakan algoritme *stoplist* atau *wordlist*. Proses *stopword removal* merupakan proses penghapusan *term* yang tidak memiliki arti atau tidak relevan. Proses ini dilakukan pada saat proses tokenisasi. Proses *filtering* yang menggunakan *stopword* adalah penghapusan kata yang ada di dalam *stopword* Bahasa Indonesia yang berisi kata-kata seperti; *ada*, *yang*, *ke*, *kepada*, dan lain sebagainya (Amin, 2013). Proses *filtering* yang menggunakan *wordlist* adalah proses penghapusan kata yang tidak ada di dalam *wordlist*. Pada Gambar 2.3 diilustrasikan proses *filtering*.





**Gambar 2.3 Ilustrasi *Filtering***

Sumber: (Merdekawati, 2014)

#### 2.2.4 Analisis Sentimen

Analisis sentimen, yang juga disebut dengan *opinion mining* merupakan bidang studi yang menganalisis pendapat, penilaian, evaluasi, sentimen, sikap dan emosi terhadap entitas seperti produk, jasa, organisasi, individu, peristiwa, topik dan atribut lainnya. Sejak awal tahun 2000, analisis sentimen telah berkembang menjadi salah satu penelitian aktif dalam *natural language processing*. Dalam beberapa tahun terakhir, analisis sentimen pada kegiatan industri juga terus berkembang (Liu, 2012). Di Amerika Serikat setidaknya terdapat 20 sampai 30 perusahaan yang menyediakan layanan untuk analisis sentimen (Liu, 2010).

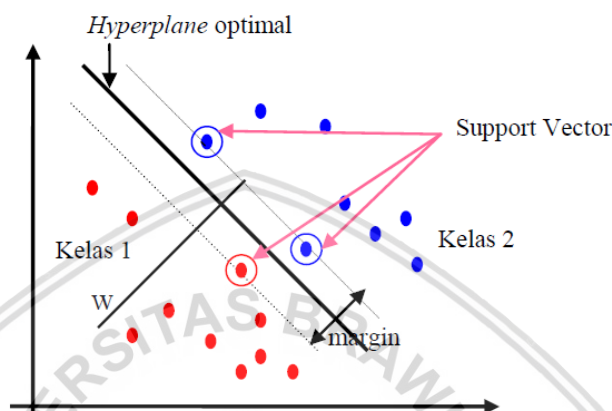
Banyaknya penelitian mengenai analisis sentimen disebabkan karena analisis sentimen memberikan manfaat yang cukup besar. Salah satu dari kegunaannya adalah dapat membantu pemantauan terhadap suatu produk dengan menggunakannya sebagai alat bantu untuk mengetahui respon masyarakat terhadap produk tersebut. Tak hanya itu analisis sentimen juga dapat digunakan untuk mengamati tanggapan masyarakat terhadap isu, sehingga dapat digunakan untuk mengetahui respon politik (Liu, 2010).

Penelitian tentang analisis sentimen terhadap dokumen Twitter telah dilakukan sebelumnya. Salah satunya adalah penelitian yang dilakukan oleh Novantirani, (2015). Analisis sentimen yang ia lakukan menggunakan metode SVM dengan objek penggunaan transportasi darat umum. Selain itu, ada juga penelitian dengan objek komentar dan *tweet* terhadap kandidat politik yang dilakukan oleh Gaurangi Patil, et al., (2014). Penelitian yang mereka lakukan juga menggunakan metode SVM. Mereka juga membandingkan metode SVM dengan metode ANN dan mengklaim bahwa metode SVM lebih baik daripada metode ANN. Penelitian ini dilakukan analisis sentimen untuk klasifikasi dokumen ke dalam dua kelas yaitu kelas sentimen positif dan kelas sentimen negatif, dengan menggunakan metode SVM kernel *Polynomial Degree 2*.

#### 2.2.5 Support Vector Machine (SVM)

SVM adalah salah satu metode yang bisa digunakan untuk pengklasifikasian. Banyak penelitian sebelumnya yang menggunakan metode ini sebagai metode

untuk mengklasifikasi. Salah satunya tentang studi kasus analisis sentimen. Klasifikasi dengan metode SVM yang dilakukan pertama kali ialah mencari *hyperplane* atau garis pembatas yang memisahkan antara suatu kelas dengan kelas lain. Untuk dapat menemukan *hyperplane* terbaik, harus diberikan maksimal *margin* yang merupakan salah satu ciri dari metode SVM. Pada Persamaan 2.1 dijelaskan cara mencari *hyperplane*. Margin adalah dua kali *hyperplane* dengan *support vector*. Pada Persamaan 2.2 dijelaskan cara mencari margin. Metode SVM dapat diilustrasikan seperti Gambar 2.4.



**Gambar 2.4 Ilustrasi SVM**  
Sumber: (Merdekawati, 2014)

$$h(x) = w \cdot x + b \quad (2.1)$$

Keterangan:

$w$  = vektor yang tegak lurus dengan *hyperplane*

$x$  = data

$b$  = nilai bias

$h(x)$  = fungsi *hyperplane*

$$\text{Margin} = |d_{h_1} - d_{h_2}| = \frac{2}{||w||} \quad (2.2)$$

Keterangan:

$d_{h_1}$  = jarak *hyperplane* kelas +1

$d_{h_2}$  = jarak *hyperplane* kelas -1

Namun pada beberapa kasus, data tidak dapat diklasifikasi menggunakan metode SVM linier. Sehingga digunakan metode SVM non-linier. Dalam metode SVM non-linier terdapat beberapa fungsi kernel. Beberapa fungsi kernel tersebut antara lain:

### 1. Polynomial

*Polynomial* kernel adalah salah satu kernel non linier. Kernel ini sangat cocok untuk digunakan pada data yang dinormalisasi (Souza, 2010). Definisi fungsinya dijelaskan pada Persamaan 2.3.

$$K(x, y) = x \cdot y \quad (2.3)$$

### 2. Gaussian

*Gaussian* kernel termasuk *radial basis function kernel*. Parameter *sigma* harus disetel hati-hati karena memegang peran penting dalam kinerja kernel ini (Souza, 2010). Definisi fungsinya dijelaskan pada Persamaan 2.4.

$$K(x, y) = \exp\left(-\frac{\|x-y\|^2}{2\sigma^2}\right) \quad (2.4)$$

### 3. Sigmoid

Kernel *sigmoid* sering disebut *Tangen Hyperbolic Tangent*. Kernel *Sigmoid* berasal dari bidang *Neural Networks*, yang mana fungsi sigmoid bipolar sering digunakan sebagai fungsi aktivasi untuk syaraf tiruan (Souza, 2010). Definisi fungsinya dijelaskan pada Persamaan 2.5.

$$K(x, y) = \tanh(\delta(x \cdot y) + c) \quad (2.5)$$

### 4. Additive

*Additive* kernel adalah salah satu kernel non linier (Merdekawati, 2014). Definisi fungsinya dijelaskan pada Persamaan 2.6.

$$K(x, y) = \sum_{i=1}^n K_i(x_i \cdot y_i) \quad (2.6)$$

Dari urian macam-macam kernel tersebut, penulis memilih kernel Polynomial dikarenakan kernel tersebut adalah kernel yang paling simpel, sederhana, dan paling mudah diimplementasikan.

## 2.2.6 Sequential Training SVM

Inti dari metode SVM adalah penentuan *hyperplane* (garis pemisah antar kelas). Vijayakumar (1999) mengembangkan solusi yang optimal untuk mempercepat proses iterasi dibandingkan dengan metode konvensional (Vijayakumar, 1999). Metode tersebut disebut *Sequential Training*. Adapun langkah-langkahnya adalah sebagai berikut.

1. Melakukan inisialisasi parameter-parameter yang digunakan, antara lain  $\lambda, \gamma, \varepsilon$ , dan  $C$  dan iterasi maksimum.
2. Setelah itu menginisialisasi  $\alpha=0$  dan menghitung matriks *Hessian*. Matriks *Hessian* dijelaskan pada Persamaan 2.7.

$$D_{ij} = y_i y_j (K(x_i, x_j) + \lambda^2) \quad (2.7)$$

Untuk  $i, j = 1, \dots, n$

Keterangan:

$x_i$  = data ke- $i$

$x_j$  = data ke- $j$

$y_i$  = kelas data ke- $i$

$y_j$  = kelas data ke- $j$

$n$  = jumlah data

$(K(x_i, x_j))$  = fungsi kernel yang digunakan

3. Mulai data ke-1 sampai ke- $n$ , hitung Persamaan 2.8, 2.9, dan 2.10.

$$E_i = \sum_{j=1}^n \alpha_j D_{ij} \quad (2.8)$$

$$\delta \alpha_i = \min\{\max[\gamma(1 - E_i), -\alpha_i], C - \alpha_i\} \quad (2.9)$$

$$\alpha_i = \alpha_i + \delta \alpha_i \quad (2.10)$$

Keterangan:

$\gamma$  = *learning rate* = konstanta  $\gamma / \max_{\{i\}} D_{ii}$

$\max_{\{i\}} D_{ii}$  = nilai maksimum dari diagonal matriks Hessien

4. Ulangi langkah ke-3 sampai iterasi maksimum.

5. Selesai.

### 2.2.7 Metode Evaluasi

Dalam hal klasifikasi teks terkadang terdapat kesalahan dalam mengklasifikasinya. Oleh sebab itu, perlu diketahui seberapa besar akurasi untuk dibandingkan dengan penelitian lain. Salah satu cara untuk mengetahui besar akurasi pengklasifikasian teks adalah menggunakan perhitungan akurasi. Akurasi adalah tingkat kedekatan antara nilai prediksi dan nilai aktual. Akurasi dijelaskan pada Persamaan 2.11:

$$\text{Akurasi} = (TP + TN) / (TP + TN + FP + FN) \quad (2.11)$$

Keterangan:

$TP$  = *True Positive*

$TN$  = *True Negative*

$FP$  = *False Positive*

$FN$  = *False Negative*

## BAB 3 METODOLOGI PENELITIAN

Dalam bab ini membahas metodologi yang digunakan dalam skripsi ini. Adapun yang dibahas dalam bab ini adalah tipe penelitian, strategi dan rancangan penelitian.

### 3.1 Tipe Penelitian

Tipe penelitian dalam skripsi ini adalah penelitian non-implementatif-analitik. Penelitian ini memfokuskan pada analisis nilai parameter yang dipakai dalam mengembangkan suatu aplikasi yang dapat membedakan mana konten radikal positif dan konten radikal negatif.

### 3.2 Strategi dan Rancangan Penelitian

Pada subbab ini dijelaskan strategi, subjek atau partisipan penelitian, lokasi penelitian, metode/teknik pengumpulan data, metode/teknik analisis data, peralatan pendukung, dan metode/teknik.

#### 3.2.1 Strategi Penelitian

Strategi penelitian yang penulis gunakan dalam penelitian ini diawali dengan pencarian literatur yang berhubungan dengan judul penelitian. Selanjutnya akan dilakukan pengumpulan dan pengolahan data. Bersamaan dengan proses pengumpulan dan pengolahan data. Kemudian dilakukan perancangan sistem yang nanti akan dijadikan pedoman dalam tahap implementasi. Setelah proses implementasi selesai, dilakukan tahap pengujian.

#### 3.2.2 Objek atau Partisipan Penelitian

Subjek penelitian dalam penelitian ini adalah *tweet* berbahasa Indonesia yang ada di Twitter. *Tweet* radikal yang dipakai berasal dari semua akun yang muncul pada halaman Twitter.

Partisipan yang terlibat dalam penelitian ini adalah Dany Ardhian, S.Pd., M.Hum, Dosen Fakultas Ilmu Budaya Universitas Brawijaya. Beliau dalam penelitian ini bertindak sebagai seorang pakar dalam menentukan data dokumen *tweet* yang dipakai termasuk konten radikal positif atau negatif. Beliau juga memberikan pengetahuan tentang konten radikal, fitur yang digunakan untuk menganalisis konten radikal dan memberikan arahan untuk melakukan survei kepada koresponden untuk menentukan suatu entitas bernilai positif atau negatif.

Pada survei yang dilakukan penulis, koresponden yang dipilih adalah teman-teman mahasiswa dari beberapa jurusan di perguruan tinggi, dengan *background* suku dan agama yang berbeda. Survei dilakukan dengan membagikan alamat Google Form.

### 3.2.3 Lokasi Penelitian

Lokasi penelitian ini terdiri dari dua lokasi. Lokasi yang pertama adalah tempat mengumpulkan data berupa dokumen *tweet* yang dipakai dalam penelitian. Lokasinya antara lain di rumah, di kampus. Lokasi yang kedua adalah lokasi wawancara dengan Bapak Dany Ardhian, S.Pd., M.Hum yang dilakukan di ruang dosen di Fakultas Ilmu Budaya Universitas Brawijaya.

### 3.2.4 Metode/Teknik Pengumpulan Data

Metode yang dipakai dalam pengumpulan data adalah mengumpulkan *tweet* yang berisi konten radikal. Pengumpulan *tweet* tersebut dilakukan dengan memasukkan *query* tertentu pada fitur pencarian di halaman Twitter. *Query* yang digunakan berupa kata yang berhubungan dengan peristiwa, pelaku, dan entitas-entitas yang berhubungan dengan konten radikal seperti agama, ras atau suku, dasar negara, dan lain-lain.

### 3.2.5 Metode/Teknik Analisis Data

Metode/teknik yang digunakan dalam menganalisis data adalah berdasarkan penjelasan yang diberikan oleh pakar. Dalam penentuan sebuah *tweet* dianggap radikal dijelaskan pada Bab 2 Landasan Kepustakaan. Untuk penentuan *tweet* tergolong radikal positif dan negatif serta fitur-fitur yang dipakai ditentukan juga oleh pakar. Fitur-fitur tersebut antara lain kata benda positif, kata benda negatif, kata kerja positif, kata kerja negatif, kata sifat positif, dan kata sifat negatif.

### 3.2.6 Peralatan Pendukung

Secara keseluruhan, peralatan pendukung yang digunakan dalam penelitian ini antara lain:

1. Kebutuhan *hardware*, meliputi:
  - Laptop/*Personal Computer* (PC) dengan Processor Intel Core i7-2630QM 2.0GHz dan memory 8GB
2. Kebutuhan *software*, meliputi:
  - *Microsoft Windows 7* sebagai sistem operasi.
  - *Netbeans 8.0.2* sebagai aplikasi untuk pengimplementasian sistem menggunakan bahasa pemrograman *Java*.
  - Data yang dibutuhkan meliputi:
    - Data dari Twitter berupa *tweet* tentang konten radikal.
    - Data kamus berupa *term* yang dipakai dalam *preprocessing* sebagai penunjang perhitungan *term frequency*.
    - Data *stopword* sebagai penunjang *text preprocessing* dalam hal *filtering*.



### 3.2.7 Metode/Teknik

Metode yang digunakan adalah mengimplementasikan algoritme *Support Vector Machine* (SVM) ke dalam bahasa pemrograman Java. Algoritme SVM yang diimplementasikan sesuai dengan manualisasi perhitungan yang dijelaskan lebih lanjut pada Bab 4 Perancangan dan Implementasi.

Metode yang digunakan untuk pengujian adalah pengujian terhadap nilai parameter-parameter. Parameter yang diuji adalah nilai parameter  $\lambda$ , nilai konstanta  $\gamma$ , dan iterasi maksimum. Pengujian yang dilakukan dijelaskan lebih lanjut pada Bab 5 Pengujian dan Analisis.

Metode yang digunakan untuk evaluasi adalah perhitungan akurasi. Perhitungan akurasi dijelaskan lebih lanjut pada Bab 2 Landasan Kepustakaan.



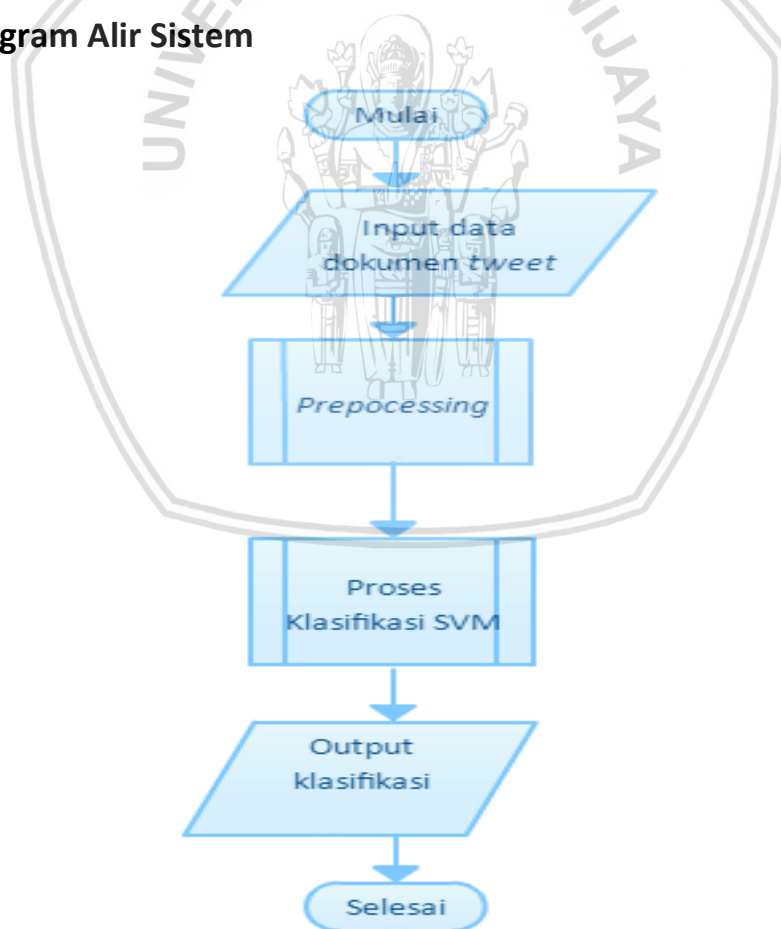
## BAB 4 PERANCANGAN DAN IMPLEMENTASI

Dalam bab ini akan membahas perancangan skripsi ini, meliputi identifikasi permasalahan, diagram alir sistem, perhitungan manual, dan perancangan antarmuka.

### 4.1 Identifikasi Permasalahan

Dalam sistem ini permasalahan yang diselesaikan adalah klasifikasi *tweet* tentang konten radikal. Data dari Twitter yang berupa *tweet* akan diinputkan ke dalam sistem untuk dilakukan *training*. Sebelum dilakukan *training*, akan dilakukan *preprocessing*. *Preprocessing* meliputi penandaan frase, tokenisasi, dan *filtering*. Untuk melakukan proses *preprocessing* dibutuhkan data berupa *feature term* yang berupa kata dan frase dari setiap kelas yang sebelumnya telah didiskusikan dengan pakar. Setelah proses, tersebut, akan dilakukan proses perhitungan *term frequency* yang kemudian akan dilakukan proses *training* dengan menggunakan metode *Sequential Training*. Setelah proses ini, barulah bisa dilakukan klasifikasi.

### 4.2 Diagram Alir Sistem

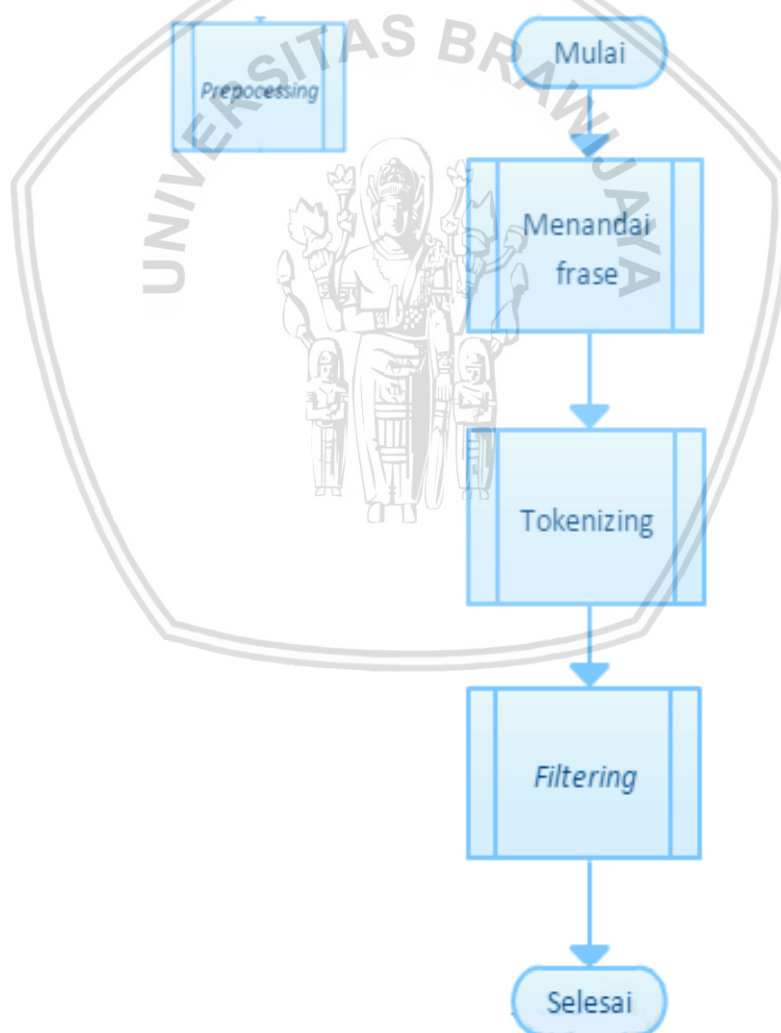


Gambar 4.1 Diagram Alir Sistem Secara Umum

Dalam perancangan sistem dibutuhkan diagram alir untuk menjelaskan proses keseluruhan sebuah sistem. Dengan dibuatnya diagram alir, diharapkan dapat membantu proses pemahaman jalannya aplikasi dengan menggunakan notasi-notasi untuk menggambarkan arus data proses sistem secara keseluruhan. Terdapat beberapa diagram alir sistem yang dapat mewakili jalannya sistem ini, antara lain diagram alir secara umum, *preprocessing*, dan metode SVM. Untuk proses secara umum dijelaskan pada Gambar 4.1.

#### 4.2.1 *Preprocessing*

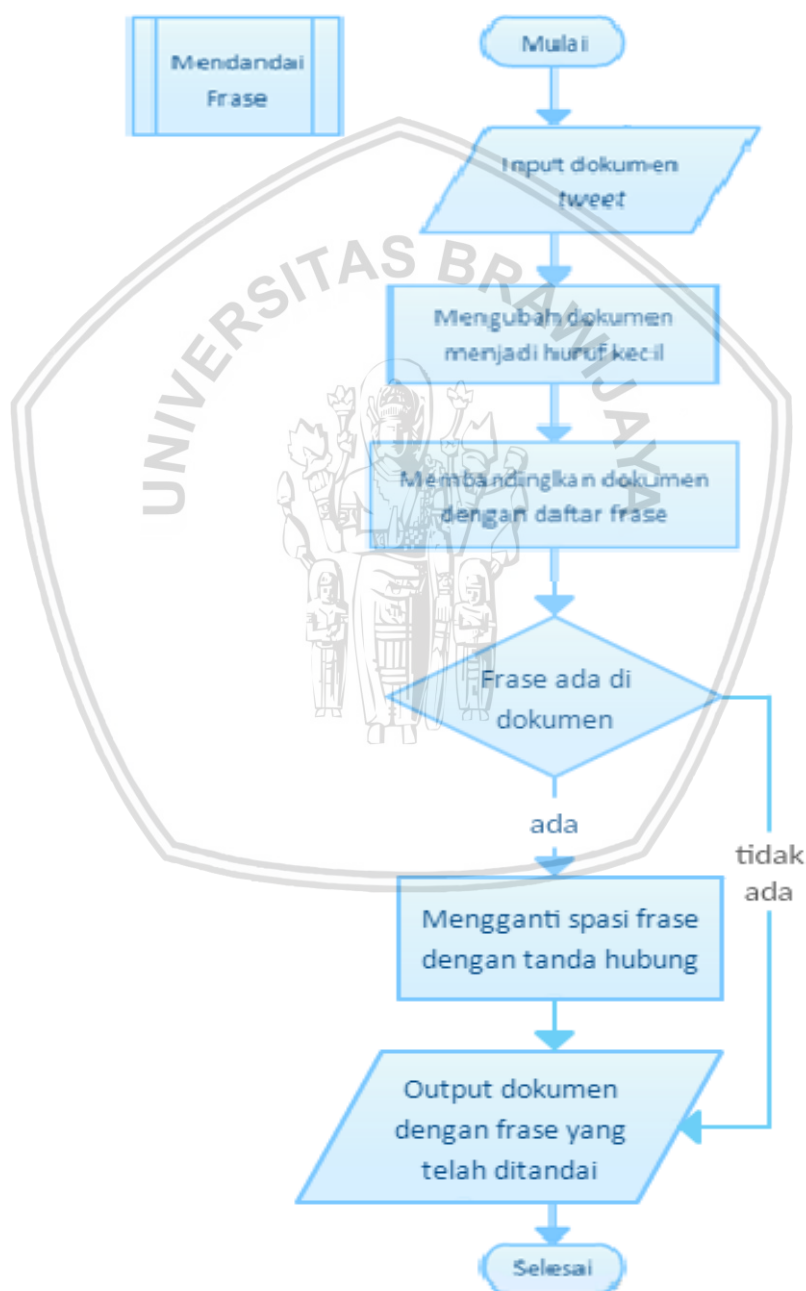
Proses yang dilakukan pada *preprocessing* adalah *tokenizing*, *filtering*, dan *stemming*. Proses *preprocessing* adalah suatu proses perubahan bentuk data yang belum terstruktur menjadi data yang terstruktur sesuai dengan kebutuhan, untuk proses *mining* yang lebih lanjut karena tidak semua kata mencerminkan isi yang terkandung dalam sebuah dokumen. Alir proses *preprocessing* ditunjukkan pada Gambar 4.2.



Gambar 4.2 Diagram Alir Sistem *Preprocessing*

#### 4.2.1.1 Penandaan Frase

Proses menandai frase merupakan proses untuk membedakan frase dengan kata-kata tunggal. Setiap kata dalam frase yang pada awalnya dipisahkan oleh spasi, dalam proses ini spasi digantikan dengan tanda hubung “-” agar dapat dibedakan dengan kata-kata tunggal dalam proses *preprocessing* selanjutnya. Penandaan frase dalam dokumen *tweet* diperlukan untuk memudahkan klasifikasi menjadi empat kelas, agar tidak ada *term* yang saling beririsan atau sama antara *term* yang terkandung dalam fitur satu dengan fitur lainnya. Gambar 4.3 menunjukkan alir proses penandaan frase.



Gambar 4.3 Diagram Alir Sistem Menandai Frase

#### 4.2.1.2 Tokenisasi

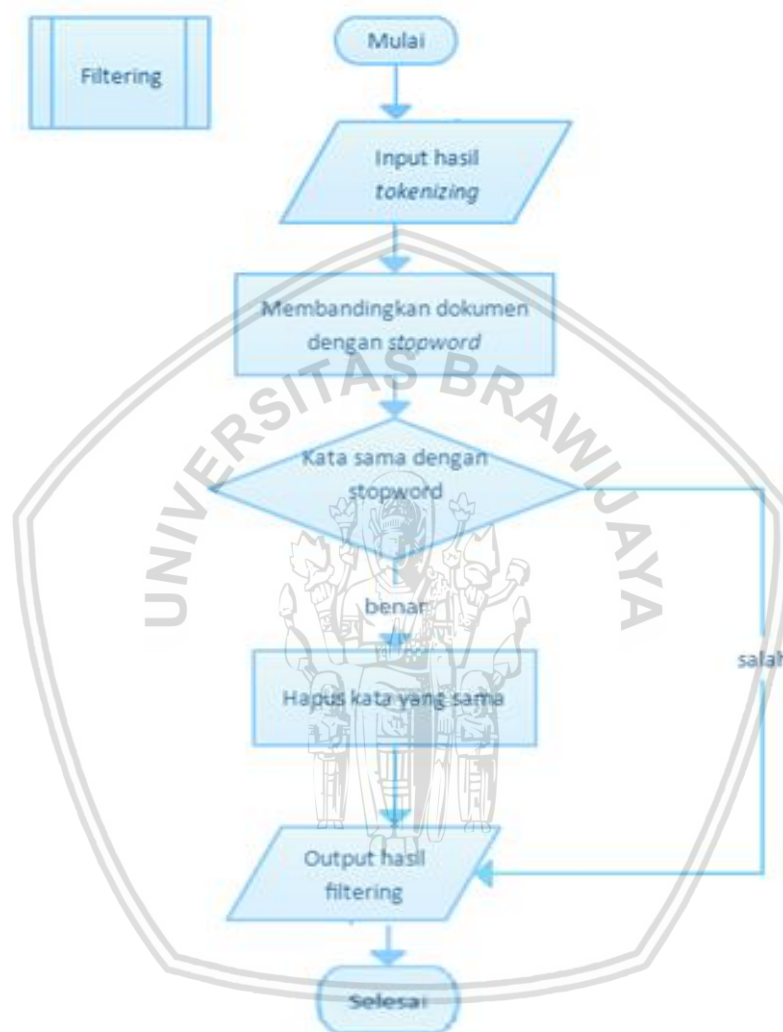
*Tokenizing* merupakan proses pemotongan *string input* berdasarkan tiap kata penyusunnya. Dengan kata lain yaitu memisahkan setiap kata yang menyusun suatu dokumen. Pada proses ini dilakukan penghilangan angka, tanda baca selain karakter “-”, dan karakter selain huruf alfabet, karena karakter-karakter tersebut dianggap sebagai pemisah kata (*delimiter*) dan tidak memiliki pengaruh terhadap pemrosesan teks. Selain itu dilakukan juga pengubahan singkatan dan kata tidak baku menjadi kata baku. Selanjutnya adalah memotong dokumen menjadi kata-kata tunggal. Alir proses *tokenizing* dapat dilihat pada Gambar 4.4.



Gambar 4.4 Diagram Alir Sistem Proses *Tokenizing*

#### 4.2.1.3 Filtering

*Filtering* adalah tahap pengambilan dari hasil *tokenizing*, yaitu kata-kata apa saja yang akan digunakan untuk merepresentasikan dokumen. Untuk mempermudah pengambilan kata penting tersebut dapat digunakan algoritme *stopword*, yaitu membuang kata-kata yang kurang penting. Alir proses *filtering* dapat dilihat pada Gambar 4.5.



Gambar 4.5 Diagram Alir Sistem Proses *Filtering*

#### 4.2.2 Klasifikasi SVM

Untuk proses klasifikasi dijelaskan pada subbab 4.3 Perhitungan Manual. Pada subbab perhitungan manual dibagi ke dalam subbab yang menjelaskan lebih rinci tahapan yang dilalui dalam perancangan sistem.

#### 4.3 Perhitungan Manual

Perhitungan manual dalam penelitian ini digunakan sebagai gambaran umum dari perancangan sistem. Pada penelitian ini menggunakan metode SVM dengan kernel *Polynomial Degree 2*. Berikut ini merupakan manualisasi setiap prosesnya.



### 4.3.1 Data Set

Data set yang digunakan pada proses manualisasi sebanyak 6 dokumen. Dokumen tersebut terdiri dari 3 dokumen kelas negatif dan 3 dokumen kelas positif. Dari 6 dokumen tersebut diambil *feature term* yang berasal dari frase dan kata untuk tiap kelas. Pada Tabel 4.1 dijelaskan data set dokumen yang digunakan pada proses manualisasi ini. Pada Tabel 4.2 dijelaskan data set *feature term* yang dipilih.

**Tabel 4.1 Tabel Data Set Dokumen Manualisasi**

No.	Dokumen	Kelas
1	Pelaku LGBT memang pantas dibunuh karena perilaku mereka bisa mendatangkan murka Allah spt halnya kaum Nabi Luth	-1
2	Ahok si anjing kafir china jelas" udah melecehkan Islam. Penghina Islam seperti Ahok pantas utk dipenggal!!!	-1
3	Para kafir yg bawa2 Islam radikal, mreka ini dkhawatirkan akan mrayu orang baik2, manusiawi dan berbudi baik mjadi terrorris. Astagfirullah.	-1
4	Pemikiran radikal (Khawarij) bisa diperbaiki dengan keadilan dan kebenaran. Oleh: Ustadz Budi Ashari, Lc	1
5	Bagi siapa saja yang anti pancasila, sebaiknya jangan tinggal di Indonesia. Pancasila masih dasar negara dan ideologi bangsa. Belum diubah	1
6	Jihad tanpa ijthad sama buruknya dengan kerja tanpa kritik.	1

**Tabel 4.2 Data Set Feature Term**

No	Jenis Fitur	List Kata	
		Kata Tunggal	Frase
1.	<b>Kata Benda Positif (KB+)</b>	islam, china, keadilan, kebenaran, ustadz, indonesia, pancasila	dasar negara, ideologi bangsa, tanpa kritik
2.	<b>Kata Kerja Positif (KK+)</b>	diperbaiki, jihad, ijthad, kerja	
3.	<b>Kata Sifat Positif (KS+)</b>	baik, manusiawi	berbudi baik
4.	<b>Kata Benda Negatif (KB-)</b>	pelaku, LGBT, anjing, kekafiran, penghina, teroris, khawarij, buruknya	murka Allah, islam radikal, pemikiran radikal, anti pancasila, penghina islam, kaum nabi luth
5.	<b>Kata Kerja Negatif (KK-)</b>	dibunuh, melecehkan, dipenggal, merayu	jangan tinggal
6.	<b>Kata Sifat Negatif (KS-)</b>	kafir	

### 4.3.2 Preprocessing

Pada manulisasi di tahap ini, dilakukan proses *preprocessing* yang meliputi proses penandaan frase, tokenisasi, *filtering* dan *stemming*. Setiap proses-proses tersebut dijelaskan pada Tabel 4.3.

**Tabel 4.3 Preprocessing**

No.	Penandaan Frase	Tokenisasi	Filtering
1	pelaku lgbt memang pantas dibunuh karena perilaku mereka bisa mendatangkan murka-alloh spt halnya kaum-nabi-luth	pelaku lgbt memang pantas dibunuh karena perilaku mereka bisa mendatangkan murka-alloh spt halnya kaum-nabi-luth	pelaku lgbt dibunuh perilaku mendatangkan murka-alloh spt halnya kaum-nabi-luth
2	ahok si anjing kafir china jelas udah melecehkan islam penghina islam seperti ahok pantas utk dipenggal	ahok si anjing kafir china jelas udah melecehkan islam penghina islam seperti ahok pantas utk dipenggal	ahok anjing kafir china jelas udah melecehkan islam penghina islam ahok pantas utk dipenggal
3	para kafir yg bawa2 islam-radikal mreka ini dkhawatirkan akan mrayu orang baik2, manusiawi dan berbudi-baik mjadi teroris astagfirullah	para kafir yg bawa2 islam-radikal mreka ini dkhawatirkan akan mrayu orang baik2, manusiawi dan berbudi-baik mjadi teroris astagfirullah	kafir bawa2 islam-radikal dkhawatirkan mrayu baik2 manusiawi berbudi-baik teroris astagfirullah
4	pemikiran-radikal khawarij bisa diperbaiki dengan keadilan dan kebenaran oleh ustadz budi ashari lc	pemikiran-radikal khawarij bisa diperbaiki dengan keadilan dan kebenaran oleh ustadz budi ashari lc	pemikiran-radikal khawarij diperbaiki keadilan kebenaran ustadz budi ashari lc
5	bagi siapa saja yang anti-pancasila sebaiknya jangan-tinggal di indonesia pancasila masih dasar-negara dan ideologi-bangsa belum diubah	bagi siapa saja yang anti-pancasila sebaiknya jangan-tinggal di indonesia pancasila masih dasar-negara dan ideologi-bangsa belum diubah	anti-pancasila jangan-tinggal indonesia pancasila dasar-negara ideologi-bangsa diubah
6	jihad tanpa ijthid sama buruknya dengan kerja tanpa-kritik	jihad tanpa ijthid sama buruknya dengan kerja tanpa-kritik	jihad tanpa ijthid sama buruknya kerja tanpa-kritik

### 4.3.3 Metode SVM

Sebelum dilakukan perhitungan, dilakukan perhitungan *term frequency* terlebih dahulu. Hasil yang didapat dapat dilihat pada Tabel 4.4.

**Tabel 4.4 Tabel Perhitungan Term Frequency**

Dokumen	KB+	KK+	KS+	KB-	KK-	KS-
1	0	0	0	4	1	0
2	2	0	0	2	2	1
3	0	0	3	2	1	1
4	3	1	0	2	0	0
5	4	0	0	1	1	0
6	1	3	0	0	0	1

Setelah didapatkan nilai *term frequency*, kemudian dilakukan proses normalisasi. Hasil normalisasi dari *term frequency* dapat dilihat pada Tabel 4.5.

**Tabel 4.5 Tabel Normalisasi *Term Frequency***

Dokumen	KB+	KK+	KS+	KB-	KK-	KS-
1	0	0	0	1	0,25	0
2	1	0	0	1	1	0,5
3	0	0	1	0,6667	0,3333	0,3333
4	1	0,3333	0	0,6667	0	0
5	1	0	0	0,25	0,25	0
6	0,3333	1	0	0	0	0,3333

Kemudian barulah dilakukan proses perhitungan SVM. Sebelum melakukannya dilakukan proses *training* dengan metode *Sequential Training SVM*. Tahap-tahap yang dilakukan antara lain:

Melakukan inisialisasi parameter-parameter yang digunakan, antara lain  $\lambda, \gamma, \varepsilon$ , dan  $C$  dan iterasi maksimum. Pada Tabel 4.6 diuraikan nilai parameter-parameter yang digunakan.

**Tabel 4.6 Parameter yang Digunakan**

$\lambda$	0,1
$\gamma$	0,1
$C$	1
$\varepsilon$	0,010
$\alpha_i$	0
threshold	0
itermax	5

1. Setelah itu menginisialisasi  $\alpha=0$  dan menghitung matriks *Hessian*.

$$D_{ij} = y_i y_j (K(x_i, x_j) + \lambda^2) \quad (4.1)$$

Untuk  $i, j = 1, \dots, n$

Keterangan:

$x_i$  = data ke- $i$

$x_j$  = data ke- $j$

$y_i$  = kelas data ke- $i$

$y_j$  = kelas data ke- $j$

$n$  = jumlah data

$(K(x_i, x_j))$  = fungsi kernel yang digunakan

Sebelum menghitung nilai matriks *Hessian*, perhitungan dari fungsi kernel yang digunakan dihitung terlebih dahulu. Tabel 4.7 menunjukkan hasil dari perhitungan kernel yang digunakan (*polynomial degree 2*).

**Tabel 4.7 Hasil Perhitungan Kernel**

id data	1	2	3	4	5	6
1	1,1289063	1,5625	0,5625375	0,4444889	0,0976563	0,1110889
2	1,5625	10,5625	1,3610722	2,7778889	2,25	0,4443556
3	0,5625375	1,3610722	2,7777778	0,1975704	0,0625	0,0493778
4	0,4444889	2,7778889	0,1975704	2,4198222	1,3611306	0,7899852
5	0,0976563	2,25	0,0625	1,3611306	1,265625	0,1735764
6	0,1110889	0,4443556	0,0493778	0,7899852	0,1735764	1,4937185

Contoh perhitungan data kolom 1 baris 1.

$$K(x, y) = (x * y)^d$$

$$K(x, y) = (x * y)^2$$

$$K(x, y) = (tf1 \text{ data1} * tf1 \text{ data1} + tf2 \text{ data1} * tf2 \text{ data1} + tf3 \text{ data1} * tf3 \text{ data1} + tf4 \text{ data1} * tf4 \text{ data1} + tf5 \text{ data1} * tf5 \text{ data1} + tf6 \text{ data1} * tf6 \text{ data1})^2$$

$$K(x, y) = (0 * 0 + 0 * 0 + 0 * 0 + 1 * 1 + 0,25 * 0,25)^2$$

$$D_{11} = 1,1289063$$

Setelah diketahui hasil dari perhitungan kernel, barulah dilakukan perhitungan matriks *Hessian*. Tabel 4.8 menunjukkan hasil dari perhitungan matriks *Hessian*.

**Tabel 4.8 Hasil Perhitungan Matriks *Hessian***

id data	1	2	3	4	5	6
1	1,1389063	1,5725	0,5725375	-0,4544889	-0,1076563	-0,1210889
2	1,5725	10,5725	1,3710722	-2,7878889	-2,26	-0,4543556
3	0,5725375	1,3710722	2,7877778	-0,2075704	-0,0725	-0,0593778
4	-0,4544889	-2,7878889	-0,2075704	2,4298222	1,3711306	0,7999852
5	-0,1076563	-2,26	-0,0725	1,3711306	1,275625	0,1835764
6	-0,1210889	-0,4543556	-0,0593778	0,7999852	0,1835764	1,5037185

Contoh perhitungan kolom 1 baris 1.

$$D_{ij} = y_i y_j (K(x_i, x_j) + \lambda^2)$$

$$D_{11} = y_1 y_1 (K(x_1, x_1) + \lambda^2)$$

$$D_{11} = 1 * 1 (tf1 \text{ data1} * tf1 \text{ data1} + tf2 \text{ data1} * tf2 \text{ data1} + tf3 \text{ data1} * tf3 \text{ data1} + tf4 \text{ data1} * tf4 \text{ data1} + tf5 \text{ data1} * tf5 \text{ data1} + tf6 \text{ data1} * tf6 \text{ data1})^2 + \lambda^2$$

$$D_{11} = 1 * 1 (0 * 0 + 0 * 0 + 0 * 0 + 1 * 1 + 0,3333 * 0,3333)^2 + 0,1^2$$

$$D_{11} = 1,1389063$$

2. Setelah hasil perhitungan matriks Hessian didapatkan, hitung Persamaan 2.8, 2.9, dan 2.10.

$$E_i = \sum_{j=1}^n \alpha_j D_{ij} \quad (4.2)$$

$$\delta \alpha_i = \min\{\max[\gamma(1 - E_i), -\alpha_i], C - \alpha_i\} \quad (4.3)$$

$$\alpha_i = \alpha_i + \delta \alpha_i \quad (4.4)$$

Keterangan:

$\gamma$  = learning rate = konstanta  $\gamma / \max_{\{i\}} D_{ii}$

$\max_{\{i\}} D_{ii}$  = nilai maksimum dari diagonal matriks Hessian

Pada perhitungan awal, iterasi dihitung dimulai dari iterasi 0. Dikarenakan nilai  $\alpha$  yang didapatkan dari hasil perhitungan masih belum diketahui. Pada iterasi ini nilai E dijelaskan pada Tabel 4.9.

**Tabel 4.9 Hasil Perhitungan E**

$\alpha_j D_{ij}$	1	2	3	4	5	6
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0
Ei	0	0	0	0	0	0

Contoh perhitungan kolom 1 baris 1

$$E_i = \sum_{j=1}^n \alpha_j D_{ij}$$

$$E_i = 0 * 1,1389603$$

$$E_i = 0$$

Setelah itu dilakukan perhitungan  $\delta \alpha$  untuk mendapatkan nilai  $\alpha$ . Pada Tabel 4.10 diuraikan hasil perhitungan  $\delta \alpha$  pada iterasi ke-0.

**Tabel 4.10 Hasil Herhitungan  $\delta \alpha_i$**

	1	2	3	4	5	6
$\delta \alpha_i$	0,1	0,1	0,1	0,1	0,1	0,1

Contoh perhitungan kolom 1 baris 1

$$\delta \alpha_i = \min\{\max[\gamma(1 - E_i), -\alpha_i], C - \alpha_i\}$$

$$\delta \alpha_i = \min\{\max[0,1(1 - 0), -0], 1 - 0\}$$

$$\delta\alpha_i = \min(0,1, 1)$$

$$\delta\alpha_i = 0,1$$

Kemudian dari hasil  $\delta\alpha$  yang didapatkan, dilakukan perhitungan untuk mendapatkan  $\alpha_i$ . Tabel 4.11 diuraikan hasil perhitungan  $\alpha_i$ .

**Tabel 4.11 Hasil Perhitungan  $\alpha_i$**

	1	2	3	4	5	6
$\alpha_i$	0,1	0,1	0,1	0,1	0,1	0,1

Contoh perhitungan kolom 1 baris 1

$$\alpha_i = \alpha_i + \delta\alpha_i$$

$$\alpha_1 = \alpha_0 + \delta\alpha_1$$

$$\alpha_i = 0 + 0,1$$

$$\alpha_i = 0,1$$

3. Ulangi langkah ke-3 sampai iterasi maksimum.

Setelah melakukan langkah ke-3, dilakukan perhitungan terus menerus sampai mencapai iterasi maksimum untuk mendapatkan nilai  $\alpha_i$  yang diperlukan untuk mencari *support vector*. Tabel 4.12 menjelaskan hasil perhitungan  $\alpha_i$  pada iterasi ke-5.

**Tabel 4.12 Hasil Perhitungan  $\alpha_i$**

	1	2	3	4	5	6
$\alpha_i$	0,3214076	0,1247767	0,2206068	0,4516632	0,5444307	0,3819127

4. Setelah proses perhitungan yang diuraikan sebelumnya, dilakukan perhitungan untuk mencari *support vector* masing-masing dokumen. Dari nilai  $\alpha_i$  terakhir yang didapat sebelumnya, diambil nilai  $\alpha_i$  terbesar dari masing-masing kelas. Tabel 4.13 menjelaskan pemilihan  $\alpha_i$  terbesar dari masing-masing kelas.

**Tabel 4.13 Pemilihan  $\alpha_i$  Terbesar**

SV	f1	f2	f3	f4	f5	f6	$\alpha_i$	kelas
1	0	0	0	1	0,25	0	0,3214076	-1
2	1	0	0	1	1	0,5	0,1247767	-1
3	0	0	1	0,6667	0,3333	0,3333	0,2206068	-1
4	1	0,3333	0	0,6667	0	0	0,4516632	1
5	1	0	0	0,25	0,25	0	0,5444307	1
6	0,3333	1	0	0,3333	0	0	0,3819127	1



Setelah itu nilai tersebut digunakan untuk menghitung fungsi kernel masing-masing kelas. Tabel 4.14 menunjukkan hasil perhitungannya.

**Tabel 4.14 Perhitungan Kernel dari  $\alpha_i$  Terbesar**

id sv	K(xi,x-)	K(xi,x+)
1	1,1289063	0,0976563
2	1,5625	2,25
3	0,5625375	0,0625
4	0,4444889	1,3611306
5	0,0976563	1,265625
6	0,1110889	0,1735764

5. Kemudian, dilakukan perhitungan  $b$  yang diperlukan dalam perhitungan *hyperplane*. Nilai  $b$  didapatkan dari Persamaan 4.5, Persamaan 4.6, Persamaan 4.7, dan Persamaan 4.8.

$$b = -\frac{1}{2} [w \cdot x^+ + w \cdot x^-] \quad (4.5)$$

$$= -\frac{1}{2} \left[ \sum_{i=1}^m \alpha_i y_i x_i \cdot x^+ + \sum_{i=1}^m \alpha_i y_i x_i \cdot x^- \right] \quad (4.6)$$

$$= -\frac{1}{2} \left[ \sum_{i=1}^m \alpha_i y_i \phi(x_i) \phi(x^+) + \sum_{i=1}^m \alpha_i y_i \phi(x_i) \phi(x^-) \right] \quad (4.7)$$

$$= -\frac{1}{2} \left[ \sum_{i=1}^m \alpha_i y_i K(x_i, x^+) + \sum_{i=1}^m \alpha_i y_i K(x_i, x^-) \right] \quad (4.8)$$

Pada Tabel 4.15 menunjukkan hasil perhitungan  $w \cdot x^+ + w \cdot x^-$

**Tabel 4.15 Hasil Perhitungan W.X**

id sv	Negatif	positif
1	-0,3628391	-0,0313875
2	-0,1949635	-0,2807475
3	-0,1240996	-0,0137879
4	0,2007593	0,6147725
5	0,0531671	0,6890451
6	0,0424263	0,066291
sigma	-0,3855496	1,0441858

Dari nilai tersebut kemudian dapat dicari nilai  $b$  dengan cara.

$$b = \frac{1}{2} * (-0,5117414 + 1,0174409)$$

$$b = -0,3293181$$

6. Pada tahap selanjutnya barulah dilakukan *testing*. *Testing* di sini data yang digunakan adalah sebanyak 4 dokumen, 3 dari kelas negatif dan 1 dari kelas positif. Dokumen tersebut diambil dari dokumen *training* nomor 1 sampai dengan 4. Dari perhitungan yang dilakukan diperoleh hasil yang dijelaskan pada Tabel 4.16, 4.17, 4.18, dan 4.19.

**Tabel 4.16 Tabel *Testing* Data 1**

SV	Kernel	$\alpha*y*k$
1	1,1289063	-0,3628391
2	1,5625	-0,1949635
3	0,5625375	-0,1240996
4	0,4444889	0,2007593
5	0,0976563	0,0531671
6	0,1110889	0,0424263
	<b>Sigma</b>	<b>-0,3855496</b>
	<b>f(x)</b>	<b>-0,7148677</b>

**Tabel 4.17 Tabel *Testing* Data 2**

SV	K	$\alpha*y*k$
1	1,5625	-0,5021994
2	10,5625	-1,3179534
3	1,3610722	-0,3002618
4	2,7778889	1,2546701
5	2,25	1,224969
6	0,4443556	0,169705
	<b>Sigma</b>	<b>0,5289295</b>
	<b>f(x)</b>	<b>0,1996114</b>

**Tabel 4.18 Tabel *Testing* Data 3**

SV	K	$\alpha*y*k$
1	0,5625375	-0,1808039
2	1,3610722	-0,16983
3	2,7777778	-0,6127967
4	0,1975704	0,0892353
5	0,0625	0,0340269
6	0,0493778	0,018858
	<b>Sigma</b>	<b>-0,8213104</b>

SV	K	$\alpha \cdot y \cdot k$
	$f(x)$	-1,1506284

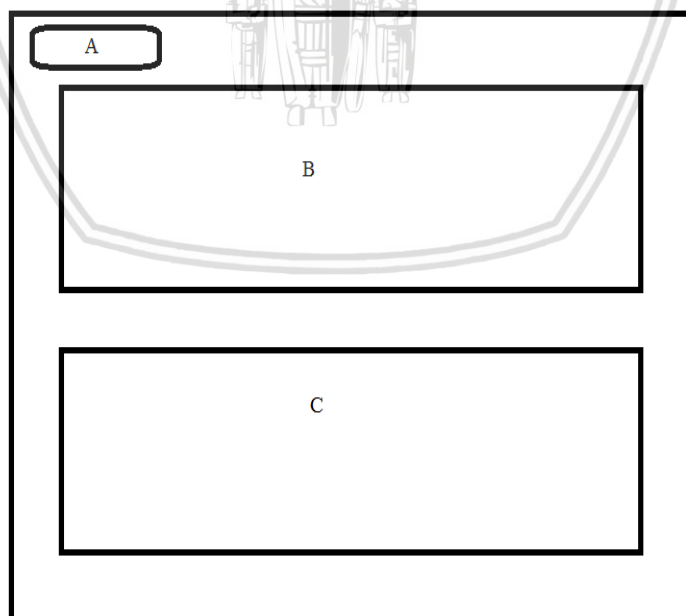
**Tabel 4.19 Tabel *Testing* Data 4**

SV	K	$\alpha \cdot y \cdot k$
1	0,4444889	-0,1428621
2	2,7778889	-0,3466157
3	0,1975704	-0,0435854
4	2,4198222	1,0929445
5	1,3611306	0,7410412
6	0,7899852	0,3017054
	<b>Sigma</b>	1,602628
	<b>f(x)</b>	1,2733099

Dari Tabel 4.16, 4.17, 4.18, dan 4.19 dijelaskan nilai  $F(x)$ . Nilai  $F(x)$  didapat dari hasil penjumlahan hasil  $\alpha_i$  dikali hasil perhitungan kernel dikali kelas ditambah b. Dari percobaan di atas hanya data kedua yang tidak mengalami kecocokan.

#### 4.4 Perancangan Antarmuka

Antarmuka dirancang untuk memudahkan pengguna untuk menggunakan sistem yang telah dibuat. Pada Gambar 4.6 dijelaskan antarmuka yang akan dirancang.



**Gambar 4.6 Perancangan Antarmuka**

Keterangan:

- A : Untuk melangkah ke proses selanjutnya
- B : Detail hasil proses yang dilakukan
- C : Detail hasil proses yang dilakukan

## 4.5 Implementasi Sistem dan Antarmuka

Pada subbab ini dijelaskan tentang pembuatan sistem dan antarmuka “Analisis Sentimen Konten Radikal pada Sosial Media Twitter Berbahasa Indonesia Menggunakan Metode *Support Vector Machine*” dengan mengacu pada perancangan yang telah dibuat sebelumnya. Sistem dirancang untuk memudahkan penelitian dalam perhitungan data yang cukup banyak. Antarmuka dirancang untuk memudahkan dalam menggunakan sistem yang telah dibuat. Pembahasan implementasi sistem dan antarmuka dijelaskan pada Subbab 4.5.1 dan Subbab 4.5.2.

### 4.5.1 Implementasi Sistem

Pada subbab ini dijabarkan proses implementasi sistem yang diuraikan dalam bentuk tabel yang berisi *source code*. Pembahasannya implementasi sistem dijelaskan pada Subbab 4.5.1.1 sampai Subbab 4.5.1.14

#### 4.5.1.1 Penandaan Frase

Penandaan frase dilakukan untuk menggabungkan dua *term* menjadi satu *term*. Proses penandaan frase pertama kali diawali dengan pencocokan dokumen *tweet* dengan untuk dicari kesamaan *term*. *Term* yang sama akan diberikan tanda hubung (-). Implementasi penandaan frase dijabarkan pada Kode Program 4.1.

1	//Penandaan Frase
2	String sqlSelectFrase = "SELECT * FROM tb_kata WHERE tipe_kata LIKE '%frase%'";
3	Statement stFrase = (Statement) jdbc.getConnection().createStatement();
4	ResultSet rsFrase = stFrase.executeQuery(sqlSelectFrase);
5	penandaanFrase = teks_case_folding;
6	while (rsFrase.next()) {
7	penandaanFrase =
8	penandaan_frase(rsFrase.getString("nama_kata"),penandaanFrase);
	}

**Kode Program 4.1 Penandaan Frase**

Penjelasan Kode Program 4.1:

- Baris 1-4 terdapat proses mengambil kalimat yang mempunyai tipe-kata frase kemudian disimpan pada *resultSet* frase.
- Baris 5 terdapat proses mengisi variable *penandaanFrase* dengan kalimat yang sudah melalui proses *casefolding*.

- Baris 6 terdapat kondisi perulangan *while* untuk mengulang setiap kata pada *resultSet* frase.
- Baris 7 berfungsi untuk menandai kata-kata yang berfungsi sebagai frase pada tabel kalimat.

#### 4.5.1.2 Tokenisasi

Pada proses ini dilakukan penghapusan kata yang mengandung karakter tidak penting, yaitu url (*http*), *@username*, dan *hashtag* (*#*). Selain itu karakter lain selain alphabet (*A-Z* dan *a-z*) dan tanda baca kecuali (*-*) akan dihapus. Implementasi tokenisasi dijabarkan pada Kode Program 4.2.

1	<code>//Tokenisasi</code>
2	<code>IndonesianSentenceTokenizer indonesianSentenceTokenizer = new IndonesianSentenceTokenizer();</code>
3	<code>ArrayList&lt;String&gt; text_tokenizing = indonesianSentenceTokenizer.tokenizeSentence(penandaanFrase);</code>

#### Kode Program 4.2 Tokenisasi

Penjelasan Kode Program 4.2:

- Baris 2 terdapat proses mempersiapkan object dengan menginstansiasi *class* *IndonesianSentenceTokenizer*.
- Baris 3 terdapat proses tokenisasi kalimat yang telah dihasilkan oleh proses penandaan frase.

#### 4.5.1.3 Filtering

Pada proses ini dilakukan penghilangan kata yang tidak penting. Kata-kata tersebut sebelumnya sudah dikumpulkan dalam *database*. Implementasi *Filtering* dijabarkan pada Kode Program 4.3.

1	<code>//Filtering</code>
2	<code>String sqlSelectTokenisasi = "SELECT * FROM tb_tokenisasi";</code>
3	<code>Statement st4 = (Statement) jdbc.getConnection().createStatement();</code>
4	<code>ResultSet rs1 = st4.executeQuery(sqlSelectTokenisasi);</code>
5	<code>while (rs1.next()) {</code>
6	<code>Object[] o1 = new Object[3];</code>
7	<code>o1[0] = rs1.getString("id_tokenisasi");</code>
8	<code>o1[1] = rs1.getString("kelas_dokumen");</code>
9	<code>o1[2] = rs1.getString("deskripsi_tokenisasi");</code>
10	<code>//remove stopwords /filtering</code>
11	<code>String text_filter = removestopword(o1[2].toString());</code>
12	<code>}</code>

#### Kode Program 4.3 Filtering

Penjelasan Kode Program 4.3:

- Baris 1-4 terdapat proses menampilkan data dari tabel tokenisasi.

- Baris 5- 9 terdapat proses memasukkan data pada tabel tokenisasi yang kemudian disimpan pada *array*.
- Baris 11 berfungsi untuk proses *remove stopwords*.

#### 4.5.1.4 Term Frequency

Pada proses ini dilakukan perhitungan *term frequency* pada setiap dokumen. *Term-term* yang ada di dalam dokumen dicocokkan dengan *term* yang sebelumnya sudah dikumpulkan dalam *database*. Implementasi perhitungan *term frequency* dijabarkan pada Kode Program 4.4.

```

1 //Term Frequency
2 Statement stFilteringKBPlus = (Statement)
  jdbc.getConnection().createStatement();
3 ResultSet rsFilteringKBPlus =
  stFilteringKBPlus.executeQuery(sqlSelectFiltering);
4 String kalimatFilterKbPlus = null;
5 int jumlahKBPlus;
6 int hitungTfKbPlus = 0;
7 while (rsFilteringKBPlus.next()) {
8   jumlahKBPlus = 0;
9   kalimatFilterKbPlus =
    rsFilteringKBPlus.getString("deskripsi_filtering");
10 //KB+
11 String sqlSelectKataKBPlus = "SELECT * FROM tb_kata WHERE
  tipe_kata LIKE '%benda%' and nilai_kata='1'";
12 Statement stKbPlus = (Statement)
  jdbc.getConnection().createStatement();
13 ResultSet rsKbPlus = stKbPlus.executeQuery(sqlSelectKataKBPlus);
14 while (rsKbPlus.next()) {
15   hitungTfKbPlus = StringUtils.countMatches(kalimatFilterKbPlus,
    rsKbPlus.getString("nama_kata"));
16   jumlahKBPlus = jumlahKBPlus + hitungTfKbPlus;
17 }
18 }

```

**Kode Program 4.4 Term Frequency**

Penjelasan Kode Program 4.4:

- Baris 2,3,7 terdapat proses menampilkan data dari tabel *filtering*.
- Baris 4,5,6 berfungsi membuat deklarasi yang kemudian akan dipakai untuk menampung data yang dihasilkan dari proses selanjutnya.
- Baris 9 berfungsi untuk menampilkan data tabel *filtering* yang nantinya akan dibandingkan dengan data kata.
- Baris 11-13 berfungsi untuk menampilkan data tabel kata yang akan dibandingkan dengan tabel *filtering*.
- Baris 14-18 berisi proses perhitungan data yang telah ditemukan sama dengan tabel kata, yang kemudian kata yang sama akan dihitung.



#### 4.5.1.5 Term Frequency (Normalisasi)

Pada proses ini dilakukan penyederhanaan nilai dari *term frequency*. Implementasi penyederhanaan nilai *term frequency* dijabarkan pada Kode Program 4.5.

```

1 //Jumlah kata maximum
2 String sqlSelectCountKKMinMax = "SELECT max(kkmin) as
  maxJumlahKata FROM tb_jumlah_kata";
3 System.out.println(sqlSelectCountKKMinMax);
4 ResultSet rsCountKKMinMax =
  stCountKKMinMax.executeQuery(sqlSelectCountKKMinMax);
5 while (rsCountKKMinMax.next()) {
6 String sqlSelectCountKKMinMin = "SELECT min(kkmin) as
  minJumlahKata FROM tb_jumlah_kata";
7 System.out.println(sqlSelectCountKKMinMin);
8 Statement stCountKKMinMin = (Statement)
  jdbc.getConnection().createStatement();
9 ResultSet rsCountKKMinMin =
  stCountKKMinMin.executeQuery(sqlSelectCountKKMinMin);
10 while (rsCountKKMinMin.next()) {
11 String sqlSelectKata = "SELECT * FROM tb_tf_jumlah_kata,
  tb_filtering WHERE tb_tf_jumlah_kata.id_filtering =
  tb_filtering.id_filtering AND tb_tf_jumlah_kata.id_filtering = "
  + i + "";
12 Statement stCountKata = (Statement)
  jdbc.getConnection().createStatement();
13 ResultSet rsCountKata = stCountKata.executeQuery(sqlSelectKata);
14 while (rsCountKata.next()) {
15 Object[] oNormalisasi = new Object[7];
16 oNormalisasi[0] = rsCountKata.getString("id_filtering");
17 oNormalisasi[1] = rsCountKata.getString("deskripsi_filtering");
18 oNormalisasi[2] = rsCountKata.getString("tipe_kata");
19 if (rsCountKata.getString("nilai_kata").equals("1")) {
20 oNormalisasi[3] = "Positif";
21 }
22 else {
23 oNormalisasi[3] = "Negatif";
24 }
25 oNormalisasi[4] = rsCountKata.getString("jumlah_kata");
26 oNormalisasi[5] = rsCountKKMinMax.getString("maxJumlahKata");
27 float x = Float.parseFloat(rsCountKata.getString("jumlah_kata"));
28 float min =
  Float.parseFloat(rsCountKKMinMin.getString("minJumlahKata"));
29 float max =
  Float.parseFloat(rsCountKKMinMax.getString("maxJumlahKata"));
30 oNormalisasi[6] = ((0.8 * (x - min)) / (max - min)) + 0.1;
31 //tblNormalisasi.addRow(oNormalisasi);
32 if ((rsCountKata.getString("nilai_kata").equals("0")) &&
  (rsCountKata.getString("tipe_kata").equals("kerja"))) {

```

```

33 String sqlUpdateKKPlusNormalisasi = "UPDATE tb_normalisasi SET
    kkmin = '" + oNormalisasi[6] + "' WHERE id = '" + oNormalisasi[0]
    + "'";
34 System.out.println(sqlUpdateKKPlusNormalisasi);
35 st.executeUpdate(sqlUpdateKKPlusNormalisasi);
36 }

```

#### Kode Program 4.5 Term Frequency (Normalisasi)

Penjelasan Kode Program 4.5:

- Baris 1-5 terdapat proses mengambil jumlah kata kerja negatif terbanyak.
- Baris 6-9 terdapat proses mengambil jumlah kata kerja negatif terendah.
- Baris 10-13 berfungsi untuk mengambil data dari tabel `tb_tf_jumlah_kata`.
- Baris 14-18 berfungsi untuk membuat objek sebagai penampung data hasil perhitungan.
- Baris 19-26 berisi kode untuk memasukkan data yang telah diambil dari tabel `tb_tf_jumlah_kata` ke dalam objek.
- Baris 27-29 berfungsi untuk mengubah 1 menjadi positif dan 0 menjadi negatif.
- Baris 30 berfungsi untuk menghitung  $((0.8 * (x - \min)) / (\max - \min)) + 0.1$ .
- Baris 31-36 berfungsi untuk memasukkan nilai dari hasil perhitungan diatas kedalam tabel `tb_normalisasi`.

##### 4.5.1.6 Proses Perhitungan Kernel

Dari nilai penyederhanaan *term frequency* dilakukukan perhitungan kernel. Implementasi Perhitungan Kernel dijabarkan pada Kode Program 4.6.

```

1 //Perhitungan Kernel
3 int c = 0;
4 tblKernel = new DefaultTableModel();
5 TblKernel.setModel(tblKernel);
6 tblKernel.addColumn("Id");
7 tblKernel.addColumn("kalimat yang di eksekusi");
8 String sqlCountKernel = "SELECT MAX(id_doc1) AS max FROM
    tb_kernel_polynomial_degree";
9 Statement stCountKernel = (Statement)
    jdbc.getConnection().createStatement();
10 ResultSet rsCountKernel =
    stCountKernel.executeQuery(sqlCountKernel);
11 while (rsCountKernel.next()) {
12     for (c = 1; c <=
        Integer.parseInt(rsCountKernel.getString("max").toString()); c++) {
13         tblKernel.addColumn(c);
14     }

```

#### Kode Program 4.6 Perhitungan Kernel

#### Penjelasan Kode Program 4.6:

- Baris 2 terdapat proses pendeklarasian variabel c dengan tipe data integer.
- Baris 3-6 terdapat proses mempersiapkan tabel yang akan menampung isi dari tabel kernel.
- Baris 7-10 berfungsi menghitung jumlah dokumen pada tabel tb\_kernel\_polinomial\_degree.
- Baris 11-14 terdapat proses perulangan dari 1 sampai jumlah dokumen untuk memasukkan nomor dokumen.

#### 4.5.1.7 Proses Perhitungan Matriks Hessian

Dari nilai kernel yang didapat maka akan dilanjutkan menuju perhitungan Matriks Hessian. Implementasi Matriks Hessian dijabarkan pada Kode Program 4.7.

```

1 //memanggil nilai dari Y yang berada di parameter
2 String sqlSelectParameterY = "SELECT y FROM tb_parameter WHERE
  ID=1";
3 Statement stSelectParameterY = (Statement)
  jdbc.getConnection().createStatement();
4 ResultSet rsSelectParameterY =
  stSelectParameterY.executeQuery(sqlSelectParameterY);
5 while (rsSelectParameterY.next()) {
6   String sqlKernelPerhitungan = "SELECT * FROM
    tb_kernel_polinomial_degree";
7   Statement stKernelPerhitungan = (Statement)
    jdbc.getConnection().createStatement();
8   ResultSet rsKernelPerhitungan =
    stKernelPerhitungan.executeQuery(sqlKernelPerhitungan);
9   while (rsKernelPerhitungan.next()) {
10    String sqlPositifNegatif = "SELECT * FROM tb_filtering WHERE
      id_filtering='" + rsKernelPerhitungan.getString("id_doc1") + "'";
11    Statement stPositifNegatif = (Statement)
      jdbc.getConnection().createStatement();
12    ResultSet rsPositifNegatif =
      stPositifNegatif.executeQuery(sqlPositifNegatif);
13    String sqlPositifNegatif2 = "SELECT * FROM tb_filtering WHERE
      id_filtering='" + rsKernelPerhitungan.getString("id_doc2") + "'";
14    Statement stPositifNegatif2 = (Statement)
      jdbc.getConnection().createStatement();
15    ResultSet rsPositifNegatif2 =
      stPositifNegatif2.executeQuery(sqlPositifNegatif2);
16    while (rsPositifNegatif.next()) {
17      rsPositifNegatif.getString("kelas_dokumen");
18    while (rsPositifNegatif2.next()) {
19      rsPositifNegatif2.getString("kelas_dokumen");
20    //=$I$5*I5*(B23+($B$13^2))
21    hasilHessian = (float)
      (Float.parseFloat(rsPositifNegatif.getString("kelas_dokumen")) *
      Float.parseFloat(rsPositifNegatif2.getString("kelas_dokumen")) *
      (Float.parseFloat(rsKernelPerhitungan.getString("nilai")) +

```

```

    (Math.pow((Float.parseFloat(rsSelectParameterY.getString("y"))),
22      2)))));
    String sqlInsertHessianMatrixPerhitungan = "INSERT INTO
    tb_matrix_hessian_perhitungan VALUES (null, ' " +
    rsKernelPerhitungan.getString("id_doc1") + "', ' " +
    rsKernelPerhitungan.getString("id_doc2") + "', ' " + hasilHessian +
    "')";
23    st.executeUpdate(sqlInsertHessianMatrixPerhitungan);
24    //System.out.println(hasilHessian);
25    }
26  }}

```

#### Kode Program 4.7 Perhitungan Matriks Hessian

Penjelasan source code 4.7:

- Baris 1-4 terdapat proses mengambil memanggil nilai dari Y yang berada di parameter.
- Baris 5-8 terdapat proses mengambil data dari tabel tb\_polinomial\_degree.
- Baris 9-15 berfungsi untuk mengambil data dari tabel tb\_tf\_jumlah\_kata.
- Baris 16-21 berisi proses perhitungan matrik hessian.
- Baris 22-26 berisi kode memasukkan hasil perhitungan matrik hessian ke dalam tabel tb\_matrix\_hessian\_perhitungan.

##### 4.5.1.8 Proses Iterasi

Setelah nilai Matriks Hessian didapat, dilakukan proses untuk perhitungan sebanyak iterasi yang telah ditentukan. Proses perhitungan iterasi dijabarkan pada Kode Program 4.8.

```

1 //proses iterasi
2 while (rsSelectIterasi.next()) {
3   for (int iterasi = 1; iterasi <=
   Integer.parseInt(rsSelectIterasi.getString("itermax").toString());
   iterasi++) {

```

#### Kode Program 4.8 Iterasi

Penjelasan Kode Program 4.8:

- Baris 2-3 berisi kode proses iterasi dari 1 sampai jumlah iterasi yang telah ditentukan.

##### 4.5.1.9 Proses Perhitungan E

Setelah iterasi ditentukan dilakukan perhitungan nilai E sebanyak iterasi yang ditentukan tadi. Proses perhitungan E dijabarkan pada Kode Program 4.9.

```

1 //menghitung E
2 String sqlKernelPerhitungan = "SELECT * FROM
  tb_matrix_hessian_perhitungan";
3 Statement stKernelPerhitungan = (Statement)
  jdbc.getConnection().createStatement();
4 ResultSet rsKernelPerhitungan =
  stKernelPerhitungan.executeQuery(sqlKernelPerhitungan);

```

```

5 while (rsKernelPerhitungan.next()) {
6 //memanggil nilai dari Y yang berada di parameter
7 String sqlSelectParameterY = "SELECT * FROM
  tb_langkah_nol_c_iterasi WHERE id='" + iterasi + "'";
8 //System.out.println(sqlSelectParameterY);
9 Statement stSelectParameterY = (Statement)
  jdbc.getConnection().createStatement();
10 ResultSet rsSelectParameterY =
  stSelectParameterY.executeQuery(sqlSelectParameterY);
11 while (rsSelectParameterY.next()) {
12 //=$I$5*I5*(B23+($B$13^2))
13 no = Integer.parseInt(rsKernelPerhitungan.getString("id_doc_2"));
14 hasilLangkahNol = (float)
  (Float.parseFloat(rsKernelPerhitungan.getString("nilai"))
15 * (Float.parseFloat(rsSelectParameterY.getString(no + 2))));
16 //System.out.println("test");
17 String sqlInsertHessianMatrixPerhitungan = "INSERT INTO
  tb_langkah_nol_a_perhitungan_iterasi VALUES (null, '" +
  rsKernelPerhitungan.getString("id_doc_1") + "', '" +
  rsKernelPerhitungan.getString("id_doc_2") + "', '" +
  hasilLangkahNol + "', '" + iterasi + "')";
18 //System.out.println(Float.parseFloat(rsSelectParameterY.getString(n
  o+2)));
19 st.executeUpdate(sqlInsertHessianMatrixPerhitungan);
20 }
21 }

```

**Kode Program 4.9 Perhitungan E**

Penjelasan Kode Program 4.9:

- Baris 2-4 terdapat proses mengambil data dari tabel tb\_matrik\_hessian.
- Baris 7-10 terdapat proses mengambil data dari tabel tb\_langkah\_nol\_c\_iterasi.
- Baris 11-15 berisi proses perhitungan nilai E.
- Baris 17-21 berfungsi untuk memasukkan hasil perhitungan nilai E kedalam tabel tb\_langkah\_no\_a\_perhitungan\_iterasi.

#### 4.5.1.10 Proses Perhitungan Delta Alfa

Setelah nilai E didapat, dilakukan perhitungan Delta Alfa yang nantinya akan dipakai untuk mencari nilai Alfa yang baru. Implementasi perhitungan Delta Alfa dijabarkan pada Kode Program 4.10

```

1 //menghitung Delta Alfa
2 String sqlCountKernel311 = "SELECT MAX(id_doc1) AS max FROM
  tb_kernel_polinomial_degree";
3 Statement stCountKernel311 = (Statement)
  jdbc.getConnection().createStatement();
4 ResultSet rsCountKernel311 =
  stCountKernel311.executeQuery(sqlCountKernel311);
5 while (rsCountKernel311.next()) {

```

```

6      for (int jumlah = 1; jumlah <=
          Integer.parseInt(rsCountKernel311.getString("max").toString())
          ; jumlah++) {
8      //mencari nilai ei dengan menjumlahkan nilai langkah a setiap
          kolom
9          String sqlSumEi = "SELECT SUM(`" + jumlah + "`) as Ei FROM
              tb_langkah_nol_a_iterasi WHERE iterasi = '" + iterasi + "'";
10         //System.out.println(sqlSumEi);
12         Statement stSumEi = (Statement)
              jdbc.getConnection().createStatement();
13         ResultSet rsSumEi = stSumEi.executeQuery(sqlSumEi);
14         while (rsSumEi.next()) {
15             //mengambil nilai y, alpha i, dan c dari parameter
16             String sqlAmbilParameter = "SELECT * FROM tb_parameter";
17             //System.out.println(sqlAmbilParameter);
18             //System.out.println(Float.parseFloat(rsSelectParameterY.getString
              (no+2)));
19             Statement stAmbilParameter = (Statement)
              jdbc.getConnection().createStatement();
20             ResultSet rsAmbilParameter =
              stAmbilParameter.executeQuery(sqlAmbilParameter);
21             while (rsAmbilParameter.next()) {
22                 //mengambil nilai y, alpha i, dan c dari parameter
23                 //System.out.println(sqlSumEi);
24                 String sqlAmbilParameterIterasi = "SELECT * FROM
                    tb_langkah_nol_c_iterasi WHERE iterasi = '" + (iterasi - 1) +
                    "'";
25                 //System.out.println(sqlAmbilParameterIterasi);
26                 Statement stAmbilParameterIterasi = (Statement)
                    jdbc.getConnection().createStatement();
27                 ResultSet rsAmbilParameterIterasi =
                    stAmbilParameterIterasi.executeQuery(sqlAmbilParameterIterasi);
28                 while (rsAmbilParameterIterasi.next()) {
29                     //MIN(MAX(y*(1-sum);-alphaI);c-alphaI)
30                     yParameter = Float.parseFloat(rsAmbilParameter.getString("y"));
31                     sum = Float.parseFloat(rsSumEi.getString("ei"));
32                     alphaI =
                    Float.parseFloat(rsAmbilParameterIterasi.getString("1"));
33                     cParameter = Float.parseFloat(rsAmbilParameterIterasi.getString("c"));
34                     // System.out.println(yParameter);
35                     // System.out.println(sum);
36                     //System.out.println(alphaI);
                    System.out.println(cParameter);
37                     hasilLangkahB = Math.min(Math.max(yParameter * (1 - sum), -
                    alphaI), cParameter - alphaI);
38                     //System.out.println(hasilLangkahB);
39                     String sqlUpdateKernel12 = "UPDATE tb_langkah_nol_b_iterasi SET
                        `" + jumlah + "` = '" + hasilLangkahB + "' WHERE id='" +
                        iterasi + "' AND iterasi = '" + iterasi + "'";
40                     //System.out.println(sqlUpdateKernel12);
41                     st.executeUpdate(sqlUpdateKernel12);

```



42	}
43	}
44	}
45	}
46	}

#### Kode Program 4.10 Perhitungan Delta Alfa

Penjelasan Kode Program 4.10:

- Baris 1-4 terdapat proses mengambil jumlah dokumen dari tabel tb\_polinomial degree.
- Baris 5-8 terdapat proses perulangan dari satu (1) sampai jumlah dokumen dari tabel tb\_polinomial\_degree.
- Baris 9-13 berfungsi untuk mengambil jumlah dari kolom jumlah yang berada pada tabel tb\_langkah\_nol\_c\_iterasi pada setiap iterasi.
- Baris 14-20 berfungsi untuk mengambil data dari tabel tb\_parameter.
- Baris 21-27 berisi kode untuk memanggil data pada tabel tb\_langkah\_nol\_c\_iterasi pada setiap iterasi.
- Baris 28-37 berfungsi untuk menghitung delta alpha dengan rumus  $\text{MIN}(\text{MAX}(y*(1-\text{sum});-\text{alpha});c-\text{alpha})$ .
- Baris 39-46 berfungsi untuk memasukkan nilai dari hasil perhitungan diatas kedalam tabel tb\_langkah\_nol\_b\_iterasi.

##### 4.5.1.11 Proses Perhitungan Alfa Baru

Setelah nilai Delta Alfa diperoleh, nilai tersebut akan digunakan untuk menghitung Alfa baru. Implementasi perhitungan Alfa baru dijabarkan pada Kode Program 4.11.

1	//memasukkan pada tabel langkah c iterasi 0
2	String sqlSelectAllJumlahKatacdcc = "SELECT * FROM
3	tb_langkah_nol_b_iterasi WHERE iterasi = '" + iterasi + "'";
4	Statement stJumlahKatacdcc = (Statement)
5	jdbc.getConnection().createStatement();
6	ResultSet rsJumlahKatacdcc =
7	stJumlahKatacdcc.executeQuery(sqlSelectAllJumlahKatacdcc);
8	while (rsJumlahKatacdcc.next()) {
9	String sqlSelectAllJumlahKatacdccc = "SELECT * FROM
10	tb_langkah_nol_b_1_iterasi WHERE iterasi = '" + iterasi + "'";
11	Statement stJumlahKatacdccc = (Statement)
12	jdbc.getConnection().createStatement();
13	ResultSet rsJumlahKatacdccc =
14	stJumlahKatacdccc.executeQuery(sqlSelectAllJumlahKatacdccc);
15	while (rsJumlahKatacdccc.next()) {
16	Object[] oJumlahKataadc = new Object[cd + 2];
17	String sqlCountKernel1cabdc = "SELECT MAX(id) AS max FROM
18	tb_langkah_nol_a";
19	Statement stCountKernel1cabdc = (Statement)
20	jdbc.getConnection().createStatement();

```

13      ResultSet rsCountKernellcabdc =
14          stCountKernellcabdc.executeQuery(sqlCountKernellcabdc);
15      while (rsCountKernellcabdc.next()) {
16          while (dad <=
17              Integer.parseInt(rsCountKernellcabdc.getString("max").toString()) +
18              1) {
19              b1 = Float.parseFloat(rsJumlahKatacdcc.getString(dad + 1));
20              //System.out.print(b1 + " ");
21              b = Float.parseFloat(rsJumlahKatacdccc.getString(dad + 1));
22              //System.out.println(b);
23              langkah_c = b + b1;
24              oJumlahKataadc[0] = rsJumlahKatacdcc.getString("iterasi");
25              oJumlahKataadc[dad - 1] = langkah_c;
26              String sqlUpdateKernell2c = "UPDATE tb_langkah_nol_c_iterasi SET
27              `" + (dad - 1) + "` = '" + langkah_c + "' WHERE iterasi = '" +
28              iterasi + "'";
29              //System.out.println(sqlUpdateKernell2c);
30              st.executeUpdate(sqlUpdateKernell2c);
31              dad++;
32          }
33      }
34  }
35  }
36  }
37  }

```

**Kode Program 4.11 Perhitungan Alfa Baru**

Penjelasan Kode Program 4.11:

- Baris 2-4 berisi kode untuk memanggil data pada tabel tb\_langkah\_nol\_b\_iterasi pada setiap iterasi.
- Baris 5-8 berisi kode untuk memanggil data pada tabel tb\_langkah\_nol\_b\_1\_iterasi pada setiap iterasi.
- Baris 9-13 terdapat proses mengambil jumlah dokumen dari tabel tb\_langkah\_nol\_a.
- Baris 14-18 berfungsi untuk mengambil data dari tabel tb\_parameter.
- Baris 21-23 berfungsi untuk menghitung alpha baru.
- Baris 24-33 berfungsi untuk memasukkan nilai dari hasil perhitungan diatas kedalam tabel tb\_langkah\_nol\_c\_iterasi.

#### **4.5.1.12 Proses Perhitungan Support Vector**

Setelah diperoleh nilai Alfa baru, nilai tersebut akan digunakan untuk menghitung nilai *Support Vector*. Implementasi perhitungan *Support Vector* dijabarkan pada Kode Program 4.12.

```

1 //memasukkan pada tabel langkah c iterasi 0
2 String sqlSelectMaxNegatif = "SELECT * FROM
  tb_normalisasi_alpha_kelas WHERE alpha_i = (SELECT MAX(alpha_i) AS
  alphaMaxPositif FROM tb_normalisasi_alpha_kelas WHERE kelas=-1)";
3 Statement statementSelectMaxNegatif = (Statement)
  jdbc.getConnection().createStatement();
4 ResultSet resultSetSelectMaxNegatif =
  statementSelectMaxNegatif.executeQuery(sqlSelectMaxNegatif);
5 while (resultSetSelectMaxNegatif.next()) {
6   String sqlSelectAllNormalisasiAlphaNegatif = "SELECT * FROM
  tb_normalisasi_alpha_kelas";
7   Statement statementSelectAllNormalisasiAlphaNegatif = (Statement)
  jdbc.getConnection().createStatement();
8   ResultSet resultSetSelectAllNormalisasiAlphaNegatif =
  statementSelectAllNormalisasiAlphaNegatif.executeQuery(sqlSelectAl
  lNormalisasiAlphaNegatif);
9   while (resultSetSelectAllNormalisasiAlphaNegatif.next()) {
10    //=(f1Terpilih*f1giliran+f2Terpilih*f2Giliran+f3Terpilih*f3Giliran+f4
    Terpilih*f4Giliran+f5Terpilih*f5Giliran+f6Terpilih*f6Giliran+alpha_te
    rpilih*alpha_giliran)^2
11    // di revisi menjadi
    =(f1Terpilih*f1giliran+f2Terpilih*f2Giliran+f3Terpilih*f3Giliran+f4Te
    rpilih*f4Giliran+f5Terpilih*f5Giliran+f6Terpilih*f6Giliran)^2
12    negatifMaxAlpha = (float)
    Math.pow(((Float.parseFloat(resultSetSelectMaxNegatif.getString("k
    bPlus")).toString()) *
    Float.parseFloat(resultSetSelectAllNormalisasiAlphaNegatif.getString(
    "kbPlus")).toString()))+
    (Float.parseFloat(resultSetSelectMaxNegatif.getString("kkPlus")).to
    String()) *
    Float.parseFloat(resultSetSelectAllNormalisasiAlphaNegatif.getString(
    "kkPlus")).toString()))+
    (Float.parseFloat(resultSetSelectMaxNegatif.getString("ksPlus")).to
    String()) *
    Float.parseFloat(resultSetSelectAllNormalisasiAlphaNegatif.getString(
    "ksPlus")).toString()))+
    (Float.parseFloat(resultSetSelectMaxNegatif.getString("kbMin")).toS
    tring()) *
    Float.parseFloat(resultSetSelectAllNormalisasiAlphaNegatif.getString(
    "kbMin")).toString()))+
    (Float.parseFloat(resultSetSelectMaxNegatif.getString("kkMin")).toS
    tring()) *
    Float.parseFloat(resultSetSelectAllNormalisasiAlphaNegatif.getString(
    "kkMin")).toString()))+
    (Float.parseFloat(resultSetSelectMaxNegatif.getString("ksMin")).toS
    tring()) *
    Float.parseFloat(resultSetSelectAllNormalisasiAlphaNegatif.getString(
    "ksMin")).toString()))), 2);
13 //H205*I205*C214
14 svNegatif = negatifMaxAlpha *
    (Float.parseFloat(resultSetSelectAllNormalisasiAlphaNegatif.getString("a
    lpha_i")).toString()) *
    (Float.parseFloat(resultSetSelectAllNormalisasiAlphaNegatif.getString("k
    elas")).toString());

```

Kode Program 4.12 Perhitungan Support Vector

#### Penjelasan Kode Program 4.12:

- Baris 1-4 terdapat proses mengambil data dengan nilai maksimal dari tabel `tb_normalisasi_alpha_kelas`.
- Baris 5-8 terdapat proses mengambil data dari tabel `tb_normalisasi_alpha_kelas`.
- Baris 9-14 berfungsi untuk perhitungan *support* vektor.

#### 4.5.1.13 Proses Perhitungan Bias(b)

```

1 //memasukkan pada tabel langkah c iterasi 0
2 String sqlSumNegatif = "SELECT SUM(negatif) AS total_negatif
3 FROM tb_sv_negatif";
4 Statement statementSumNegatif = (Statement)
5 jdbc.getConnection().createStatement();
6 ResultSet resultSetSumNegatif =
7 statementSumNegatif.executeQuery(sqlSumNegatif);
8 while (resultSetSumNegatif.next()) {
9     lblSumPositif.setText(resultSetSumPositif.getString("total_positif"));
10    total_positif =
11    Float.parseFloat(resultSetSumPositif.getString("total_positif").
12    toString());
13    lblSumNegatif.setText(resultSetSumNegatif.getString("total_negatif"));
14    total_negatif =
15    Float.parseFloat(resultSetSumNegatif.getString("total_negatif").
16    toString());
17    total_b = (float) (-0.5 * (total_positif + total_negatif));
18    lblB.setText(" jadi nilai b adalah= " + total_b);
19 }

```

**Kode Program 4.13 Perhitungan Bias**

#### Penjelasan Kode Program 4.13:

- Baris 1-4 terdapat proses mengambil jumlah negatif dari `tb_sv_negatif`.
- Baris 5-10 terdapat proses perhitungan bias(b).
- Baris 11-12 berfungsi untuk menampilkan ke layar nilai bias(b).

#### 4.5.1.14 Proses Testing

```

1 String sqlSelectMaxPositif = "SELECT * FROM
2 tb_normalisasi_alpha_kelas";
3 Statement statementSelectMaxPositif = (Statement)
4 jdbc.getConnection().createStatement();
5 ResultSet resultSetSelectMaxPositif =
6 statementSelectMaxPositif.executeQuery(sqlSelectMaxPositif);
7 while (resultSetSelectMaxPositif.next()) {
8     String sqlSelectAllNormalisasiAlphaPositifNegatif = "SELECT *
9 FROM tb_normalisasi_alpha_kelas";
10    Statement statementSelectAllNormalisasiAlphaPositifNegatif =
11    (Statement) jdbc.getConnection().createStatement();

```

7	ResultSet resultSelectAllNormalisasiAlphaPositifNegatif = statementSelectAllNormalisasiAlphaPositifNegatif.executeQuery(sqlSelectAllNormalisasiAlphaPositifNegatif);
8	while (resultSelectAllNormalisasiAlphaPositifNegatif.next()) {
9	// =(f1Terpilih*f1giliran+f2Terpilih*f2Giliran+f3Terpilih*f3Giliran+f4 Terpilih*f4Giliran+f5Terpilih*f5Giliran+f6Terpilih*f6Giliran+alpha_ terpilih*alpha_giliran)^2
10	positifMaxAlpha = (float) Math.pow(((Float.parseFloat(resultSetSelectMaxPositif.getString("kbPlus")).toString()) * Float.parseFloat(resultSelectAllNormalisasiAlphaPositifNegatif.getString("kbPlus")).toString())) + (Float.parseFloat(resultSetSelectMaxPositif.getString("kkPlus")).toString()) * Float.parseFloat(resultSelectAllNormalisasiAlphaPositifNegatif.getString("kkPlus")).toString()) + (Float.parseFloat(resultSetSelectMaxPositif.getString("ksPlus")).toString()) * Float.parseFloat(resultSelectAllNormalisasiAlphaPositifNegatif.getString("ksPlus")).toString()) + (Float.parseFloat(resultSetSelectMaxPositif.getString("kbMin")).toString()) * Float.parseFloat(resultSelectAllNormalisasiAlphaPositifNegatif.getString("kbMin")).toString()) + (Float.parseFloat(resultSetSelectMaxPositif.getString("kkMin")).toString()) * Float.parseFloat(resultSelectAllNormalisasiAlphaPositifNegatif.getString("kkMin")).toString()) + (Float.parseFloat(resultSetSelectMaxPositif.getString("ksMin")).toString()) * Float.parseFloat(resultSelectAllNormalisasiAlphaPositifNegatif.getString("ksMin")).toString()), 2);
11	//=H205*I205*C214
12	svPositif = positifMaxAlpha * (Float.parseFloat(resultSelectAllNormalisasiAlphaPositifNegatif.getString("alpha_i")).toString()) * (Float.parseFloat(resultSelectAllNormalisasiAlphaPositifNegatif.getString("kelas")).toString());

**Kode Program 4.14 Testing**

Penjelasan Kode Program 4.14:

- Baris 1-3 terdapat proses mengambil data dari tabel tb\_normalisasi\_alpha\_kelas.
- Baris 5-7 terdapat proses mengambil data dari tabel tb\_normalisasi\_alpha\_kelas.
- Baris 8-12 terdapat proses perhitungan *testing*.

#### 4.5.2 Implementasi Antarmuka

Pada subbab ini dijelaskan implementasi antarmuka sistem yang dibuat. Subbab ini dibagi menjadi 2 subbab yang berisi tentang implementasi antarmuka *preprocessing* dan implementasi antarmuka metode SVM.

#### 4.5.2.1 Antarmuka Halaman *Preprocessing*

Halaman *preprocessing* dan pelatihan berfungsi sebagai halaman untuk menjalankan proses *preprocessing* dan pelatihan. Dalam halaman ini terdapat form untuk memasukkan data dan tombol navigasi untuk memulai proses *preprocessing* dan pelatihan. Selain itu, terdapat tabel yang menampilkan hasil dari setiap tahap *preprocessing* dan pelatihan. Tampilan halaman *preprocessing* dan pelatihan dapat dilihat pada Gambar 4.7 sampai Gambar 4.12. Gambar 4.7 dimuat pada halaman selanjutnya.

The screenshot shows a web application interface for document preprocessing. It includes a 'Lihat Normalisasi Term' button, a text area for 'isi dokumen', and radio buttons for 'kelas' (positif/negatif). Below are three tables: 'DOKUMEN', 'PENANDAAN FRASE', 'TOKENISASI', and 'FILTERING'. The 'DOKUMEN' table lists document IDs, classes, and descriptions. The 'PENANDAAN FRASE' table shows phrase annotations. The 'TOKENISASI' table shows tokenization results. The 'FILTERING' table shows filtered results. A 'Master Reset' button is in the top right.

Id Dokumen	Kelas	Deskripsi Dokumen
1	-1	Pelaku LGBT meman...
2	-1	Ahok si anjing kafir chi...
3	-1	Para kafir yg bawa2 Is...
4	1	Pemikiran radikal (Kh...
5	1	Bagi siapa saja yang ...
6	1	Jihad tanpa ijihad sa...

Id Penandaan...	Kelas	Deskripsi Pe...
1	-1	pelaku lgbt m...
2	-1	ahok si anjin...
3	-1	para kafir yg b...
4	1	pemikiran-ra...
5	1	bagi siapa sa...
6	1	jihad tanpa jti...

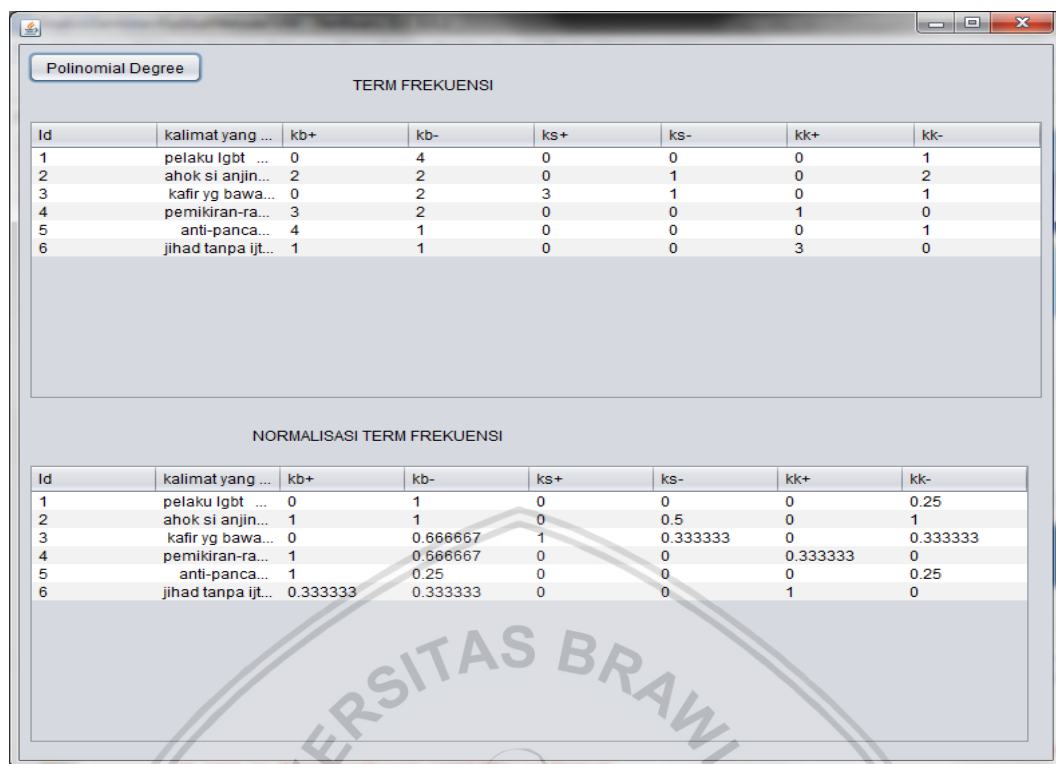
Id Tokenisasi	Kelas	Deskripsi To...
1	-1	pelaku
1	-1	lgbt
1	-1	memang
1	-1	pantas
1	-1	dibunuh
1	-1	karena
1	-1	perilaku
1	-1	mereka
1	-1	bisa
1	-1	mendatang...
1	-1	murka-allah
1	-1	spt
1	-1	halnya
1	-1	kaum-nabi-l...
2	-1	ahok
2	-1	si
2	-1	anjing
2	-1	kafir
2	-1	china
2	-1	jelas
2	-1	udah

Id Filtering	Kelas	Deskripsi Filt...
1	-1	pelaku lgbt d...
2	-1	ahok si anjin...
3	-1	kafir yg bawa...
4	1	pemikiran-ra...
5	1	anti-panca...
6	1	jihad tanpa jti...

Gambar 4.7 Antarmuka Halaman *Preprocessing* (1)

Pada Gambar 4.7 menampilkan antarmuka halaman *preprocessing*. Pada antarmuka *preprocessing* diuraikan proses penandaan frase, tokenisasi, dan proses *filtering*. Pada halaman tersebut terdapat tombol untuk melangkah ke proses selanjutnya.





Polinomial Degree

TERM FREKUENSI

Id	kalimat yang ...	kb+	kb-	ks+	ks-	kk+	kk-
1	pelaku lgbt ...	0	4	0	0	0	1
2	ahok si anjin...	2	2	0	1	0	2
3	kafir yg bawa...	0	2	3	1	0	1
4	pemikiran-ra...	3	2	0	0	1	0
5	anti-panca...	4	1	0	0	0	1
6	jihad tanpa ijt...	1	1	0	0	3	0

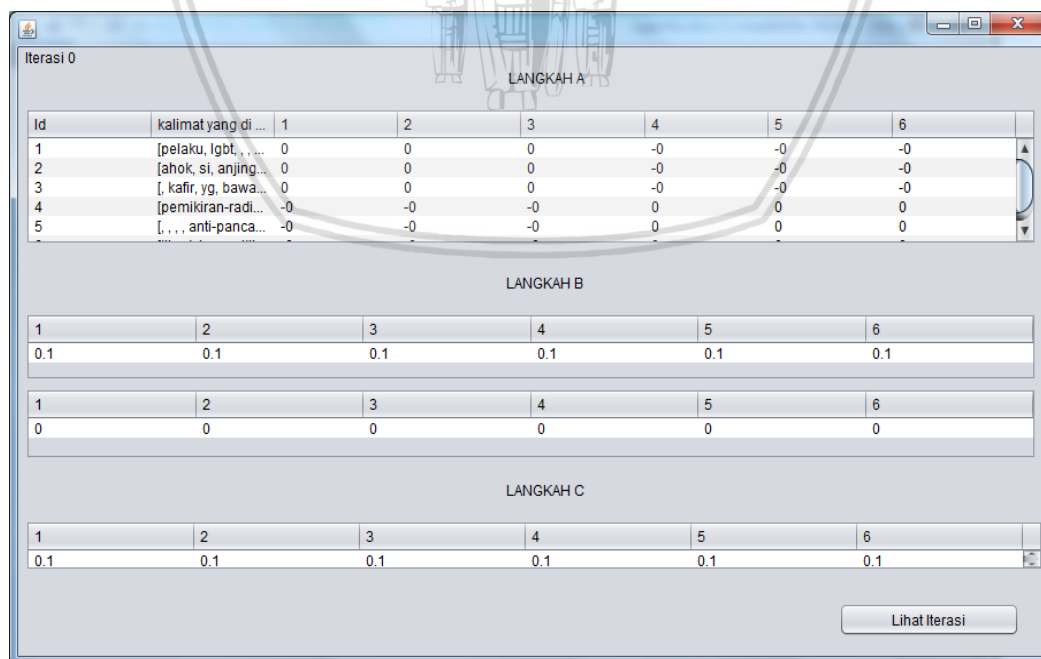
NORMALISASI TERM FREKUENSI

Id	kalimat yang ...	kb+	kb-	ks+	ks-	kk+	kk-
1	pelaku lgbt ...	0	1	0	0	0	0.25
2	ahok si anjin...	1	1	0	0.5	0	1
3	kafir yg bawa...	0	0.666667	1	0.333333	0	0.333333
4	pemikiran-ra...	1	0.666667	0	0	0.333333	0
5	anti-panca...	1	0.25	0	0	0	0.25
6	jihad tanpa ijt...	0.333333	0.333333	0	0	1	0

Gambar 4.8 Antarmuka Preprocecing (2)

Gambar 4.8 merupakan tampilan antarmuka halaman prepocecing perhitungan *term frequency*. Pada halaman tersebut menampilkan form perhitungan *term frequency* dan tombol untuk proses selanjutnya.

#### 4.5.2.2 Implementasi Metode SVM



Iterasi 0

LANGKAH A

Id	kalimat yang di ...	1	2	3	4	5	6
1	[pelaku, lgbt, ...	0	0	0	-0	-0	-0
2	[ahok, si, anjing...	0	0	0	-0	-0	-0
3	[, kafir, yg, bawa...	0	0	0	-0	-0	-0
4	[pemikiran-radi...	-0	-0	-0	0	0	0
5	[, ..., anti-panca...	-0	-0	-0	0	0	0

LANGKAH B

1	2	3	4	5	6
0.1	0.1	0.1	0.1	0.1	0.1

1	2	3	4	5	6
0	0	0	0	0	0

LANGKAH C

1	2	3	4	5	6
0.1	0.1	0.1	0.1	0.1	0.1

Lihat Iterasi

Gambar 4.9 Antarmuka Pelatihan (1)

Gambar 4.9 merupakan antarmuka halaman pelatihan yang menampilkan perhitungan iterasi 0. Tombol “Lihat Iterasi” digunakan untuk melihat perhitungan iterasi selanjutnya.

iterasi 1+

LANGKAH A

Id	kalimat yang ...	Iterasi Ke	1	2	3	4	5	6
1	pelaku lgbt ...	1	0.113891	0.15725	0.05725	-0.0454445	-0.0107656	-0.0121111
1	pelaku lgbt ...	2	0.198162	0.188481	0.0893558	-0.0856574	-0.0211111	-0.0219785
1	pelaku lgbt ...	3	0.260518	0.194685	0.107361	-0.121241	-0.0310529	-0.0300178
1	pelaku lgbt ...	4	0.306656	0.195916	0.117458	-0.152728	-0.0406069	-0.0365678
1	pelaku lgbt ...	5	0.340795	0.196162	0.123121	-0.18059	-0.0497881	-0.0419043

LANGKAH B

Iterasi ke	1	2	3	4	5	6
1	0.073993	0.0198611	0.0560802	0.0884877	0.0960981	0.0814737
2	0.0547497	0.00394481	0.03145	0.0783008	0.0923484	0.0663797

Iterasi ke	1	2	3	4	5	6
1	0.1	0.1	0.1	0.1	0.1	0.1
2	0.173993	0.119861	0.15608	0.188488	0.196098	0.181474

LANGKAH C

Iterasi ke	1	2	3	4	5	6
0	0.1	0.1	0.1	0.1	0.1	0.1
1	0.173993	0.119861	0.15608	0.188488	0.196098	0.181474

Menuju Pengetesan

Gambar 4.10 Antarmuka Halaman Pelatihan (2)

Gambar 4.10 merupakan antarmuka halaman pelatihan yang menampilkan hasil perhitungan iterasi 1 sampai iterasi maksimum. Tombol Menuju Pengetesan digunakan untuk menampilkan proses perhitungan *support vector* dan bias.

TERM FREKUENSI

id	kb+	kk+	ks+	kb-	kk-	ks-	Alpha I	Kelas
1	0	0	0	1	0.25	0	0.321409	-1
2	1	0	0	1	1	0.5	0.124775	-1
3	0	0	1	0.666667	0.333333	0.333333	0.220605	-1
4	1	0.333333	0	0.666667	0	0	0.451637	1
5	1	0	0	0.25	0.25	0	0.544429	1
6	0.333333	1	0	0.333333	0	0	0.381898	1

Testing Machine

K(xi,x-)		K(xi,x+)		positif		negatif	
id sv	K(xi,x-)	id sv	K(xi,x+)	id sv	Positif	id sv	Negatif
1	1.1289062	1	0.09765625	1	-0.031387...	1	-0.362840...
2	1.5625	2	2.25	2	-0.280743...	2	-0.194960...
3	0.56250036	3	0.0625	3	-0.013787...	3	-0.124090...
4	0.44444486	4	1.3611113	4	0.6147282	4	0.20072775
5	0.09765625	5	1.265625	5	0.689043	5	0.053166...

1.0441536391153932      -0.38556428253650665

testing

jadi nilai b adalah : -0.3292947

Gambar 4.11 Antarmuka Halaman Pelatihan (3)

Gambar 4.11 merupakan halaman pelatihan yang perhitungan *support vector* dan perhitungan bias. Tombol *testing* digunakan untuk menuju ke proses *testing*.

#### 4.5.2.3 Antarmuka Halaman Pengujian

Halaman pengujian sebagai halaman untuk menampilkan hasil pengujian. Dalam halaman ini terdapat perhitungan pengujian dan akurasi sistem. Tampilan halaman pengujian dapat dilihat Gambar 5.6.

Id sv	Testing ke N atau Id Doc	k	Alpha*y*k
1	1	1.1289062	-0.36284062
2	1	1.5625	-0.19496094
3	1	0.56250036	-0.12409039
4	1	0.44444486	0.20072775
5	1	0.09765625	0.053166896
6	1	0.11111088	0.042433023
1	2	1.5625	-0.50220156
2	2	10.5625	-1.317936
3	2	1.3611107	-0.30026782
4	2	2.7777789	1.2545477
5	2	2.25	1.2249652
6	2	0.44444352	0.1697321
1	3	0.56250036	-0.18079267
2	3	1.3611107	-0.16983259
3	3	2.7777781	-0.6127918
4	3	0.19753124	0.08921242
5	3	0.0525	0.034025813
6	3	0.049382657	0.018859137

Id Doc	Sigma	f(x)
1	-0.38556428253650665	-0.714859
2	0.5288397073745728	0.199545
3	-0.8213186711072922	-1.1506134
4	1.6026026122272015	1.2733079
5	1.0441536391153932	0.714859
6	0.9197947522625327	0.5905

Id Doc	Kelas Dokumen	Hasil Sistem
1	-1	-0.714859
2	-1	0.199545
3	-1	-1.1506134
4	1	1.2733079
5	1	0.714859
6	1	0.5905

**Akurasi : 83 %**

Gambar 4.12 Antarmuka Halaman Pengujian

Gambar 4.12 merupakan halaman pengujian dan perhitungan akurasi sistem. Pada antarmuka ini, semua proses penelitian selesai dilakukan.

## 4.6 Skenario Pengujian

Pengujian diperlukan dalam penelitian ini untuk menunjukkan bahwa perangkat lunak yang dibuat telah mampu bekerja sesuai dengan kebutuhan yang diinginkan. Pengujian yang dilakukan adalah sebagai berikut.

1. Pengujian tingkat akurasi dipengaruhi oleh nilai parameter  $\lambda$ . Hasil pengujian akan ditampilkan ke dalam Tabel 4.20.

Tabel 4.20 Tabel Pengujian 1

Percobaan	$\lambda$	$\gamma$	C	$\varepsilon$	Iterasi Maksimum	Akurasi (%)

Percobaan	$\lambda$	$\gamma$	C	$\varepsilon$	Iterasi Maksimum	Akurasi (%)

2. Pengujian tingkat akurasi dipengaruhi oleh nilai konstanta ( $\gamma$ ). Hasil pengujian akan ditampilkan ke dalam Tabel 4.21.

**Tabel 4.21 Tabel Pengujian 2**

Percobaan	$\lambda$	$\gamma$	C	$\varepsilon$	Iterasi Maksimum	Akurasi (%)

3. Perancangan pengujian tingkat akurasi dipengaruhi oleh jumlah iterasi dalam melakukan *training*. Hasil pengujian akan ditampilkan ke dalam Tabel 4.22.

**Tabel 4.22 Tabel Pengujian 3**

Percobaan	$\lambda$	$\gamma$	C	$\varepsilon$	Iterasi Maksimum	Akurasi (%)

## BAB 5 PENGUJIAN DAN ANALISIS

Bab ini membahas tentang pengujian, hasil pengujian dan analisis pengujian sistem. Pengujian, hasil pengujian dan analisis pengujian sistem diuraikan pada Subbab 5.1, 5.2, dan 5.3.

### 5.1 Pengujian

Pada pengujian digunakan dokumen Twitter berupa *tweet* radikal berbahasa Indonesia yang diambil secara manual. Dari data tersebut kemudian dikonsultasikan kepada Dosen Fakultas Ilmu Bahasa untuk diketahui kelas dari dokumen-dokumen tersebut. Dokumen yang digunakan dalam pengujian ini sebanyak 80 dokumen (60 dokumen negatif dan 20 dokumen positif) untuk data latih dan 20 dokumen (15 dokumen negatif dan 5 dokumen positif) untuk data uji. Pengujian yang dilakukan antara lain:

1. Pengujian terhadap nilai parameter  $\lambda$ .
2. Pengujian terhadap nilai konstanta  $\gamma$ .
3. Pengujian terhadap iterasi maksimum.

### 5.2 Hasil Pengujian

Pada subbab ini diuraikan hasil pengujian sistem yang akan dijabarkan pada subbab 5.2.1, 5.2.2, dan 5.2.3.

#### 5.2.1 Hasil Pengujian Terhadap Nilai Parameter $\lambda$

Pengujian ini dilakukan untuk mencari akurasi terbesar dengan menggunakan nilai parameter  $\lambda$  terkecil. Nilai parameter yang digunakan dalam pengujian ini adalah 0,1, 0,5, 1, 1,5, dan 2. Hasil pengujian nilai parameter  $\lambda$  terhadap sistem dijelaskan pada Tabel 5.1. Dokumen yang relevan pada tiap percobaan dapat dilihat pada Tabel 5.2.

**Tabel 5.1 Hasil Pengujian Terhadap Nilai Parameter  $\lambda$**

Percobaan	$\lambda$	$\gamma$	C	$\varepsilon$	Iterasi Maksimum	Akurasi (%)
1	0,1	0,1	1	0,001	1	70
2	0,5	0,1	1	0,001	1	70
3	1	0,1	1	0,001	1	70
4	1,5	0,1	1	0,001	1	70
5	2	0,1	1	0,001	1	70

Tabel 5.2 Dokumen Relevan Pada Pengujian Terhadap Nilai Parameter  $\lambda$ 

No.	Nomor Tweet	Prediksi Kelas	Cocok (Ya/Tidak)				
			Percobaan				
			1	2	3	4	5
1	1	Negatif	Ya	Ya	Ya	Ya	Ya
2	2	Negatif	Tidak	Tidak	Tidak	Tidak	Tidak
3	3	Negatif	Ya	Ya	Ya	Ya	Ya
4	4	Negatif	Ya	Ya	Ya	Ya	Ya
5	5	Negatif	Ya	Ya	Ya	Ya	Ya
6	6	Negatif	Tidak	Tidak	Tidak	Tidak	Tidak
7	7	Negatif	Ya	Ya	Ya	Ya	Ya
8	8	Negatif	Ya	Ya	Ya	Ya	Ya
9	9	Negatif	Ya	Ya	Ya	Ya	Ya
10	10	Negatif	Tidak	Tidak	Tidak	Tidak	Tidak
11	11	Negatif	Tidak	Tidak	Tidak	Tidak	Tidak
12	12	Positif	Tidak	Tidak	Tidak	Tidak	Tidak
13	13	Positif	Ya	Ya	Ya	Ya	Ya
14	14	Positif	Ya	Ya	Ya	Ya	Ya
15	15	Negatif	Ya	Ya	Ya	Ya	Ya
16	16	Positif	Ya	Ya	Ya	Ya	Ya
17	17	Negatif	Tidak	Tidak	Tidak	Tidak	Tidak
18	18	Positif	Ya	Ya	Ya	Ya	Ya
19	19	Negatif	Ya	Ya	Ya	Ya	Ya
20	20	Negatif	Ya	Ya	Ya	Ya	Ya

### 5.2.2 Hasil Pengujian Terhadap Nilai Konstanta $\gamma$

Pengujian ini dilakukan untuk mencari akurasi terbesar dengan menggunakan nilai konstanta  $\gamma$ . Nilai konstanta  $\gamma$  yang digunakan dalam pengujian ini adalah 0,1, 0,01, 0,001, 0,0001, dan 0,00001. Hasil pengujian nilai konstanta  $\gamma$  terhadap sistem dijelaskan pada Tabel 5.3. Dokumen yang relevan pada tiap percobaan dapat dilihat pada Tabel 5.4.

Tabel 5.3 Hasil Pengujian Terhadap Nilai Konstanta  $\gamma$ 

Percobaan	$\lambda$	$\gamma$	C	$\epsilon$	Iterasi Maksimum	Akurasi (%)
1	0,1	0,1	1	0,001	5	70
2	0,1	0,01	1	0,001	5	70
3	0,1	0,001	1	0,001	5	65



Percobaan	$\lambda$	$\gamma$	C	$\varepsilon$	Iterasi Maksimum	Akurasi (%)
4	0,1	0,0001	1	0,001	5	45
5	0,1	0,00001	1	0,001	5	40

**Tabel 5.4 Dokumen Relevan Pada Pengujian Terhadap Nilai Konstanta  $\gamma$**

No.	Nomor Tweet	Prediksi Kelas	Cocok (Ya/Tidak)				
			Percobaan				
			1	2	3	4	5
1	1	Negatif	Ya	Ya	Ya	Ya	Ya
2	2	Negatif	Tidak	Tidak	Tidak	Ya	Ya
3	3	Negatif	Ya	Ya	Ya	Tidak	Tidak
4	4	Negatif	Ya	Ya	Ya	Tidak	Tidak
5	5	Negatif	Ya	Ya	Ya	Tidak	Tidak
6	6	Negatif	Tidak	Tidak	Tidak	Ya	Ya
7	7	Negatif	Ya	Ya	Ya	Tidak	Tidak
8	8	Negatif	Ya	Ya	Ya	Tidak	Tidak
9	9	Negatif	Ya	Ya	Ya	Tidak	Tidak
10	10	Negatif	Tidak	Tidak	Tidak	Tidak	Tidak
11	11	Negatif	Tidak	Tidak	Tidak	Tidak	Tidak
12	12	Positif	Tidak	Tidak	Tidak	Ya	Ya
13	13	Positif	Ya	Ya	Tidak	Ya	Ya
14	14	Positif	Ya	Ya	Ya	Ya	Ya
15	15	Negatif	Ya	Ya	Ya	Tidak	Tidak
16	16	Positif	Ya	Ya	Ya	Ya	Ya
17	17	Negatif	Tidak	Tidak	Tidak	Tidak	Tidak
18	18	Positif	Ya	Ya	Ya	Ya	Ya
19	19	Negatif	Ya	Ya	Ya	Ya	Tidak
20	20	Negatif	Ya	Ya	Ya	Tidak	Tidak

### 5.2.3 Hasil Pengujian Terhadap Iterasi Maksimum

Pengujian ini dilakukan untuk mencari akurasi terbesar dengan menggunakan iterasi maksimum. Iterasi yang digunakan dalam pengujian ini adalah 1, 2, 3, 4, dan 5. Hasil pengujian iterasi maksimum terhadap sistem dijelaskan pada Tabel 5.5. Dokumen yang relevan pada tiap percobaan dapat dilihat pada Tabel 5.6.

Tabel 5.5 Hasil Pengujian Terhadap Iterasi Maksimum

Percobaan	$\lambda$	$\gamma$	C	$\varepsilon$	Iterasi Maksimum	Akurasi (%)
1	0,1	0,1	1	0,001	1	70
2	0,1	0,1	1	0,001	2	70
3	0,1	0,1	1	0,001	3	70
4	0,1	0,1	1	0,001	4	70
5	0,1	0,1	1	0,001	5	70

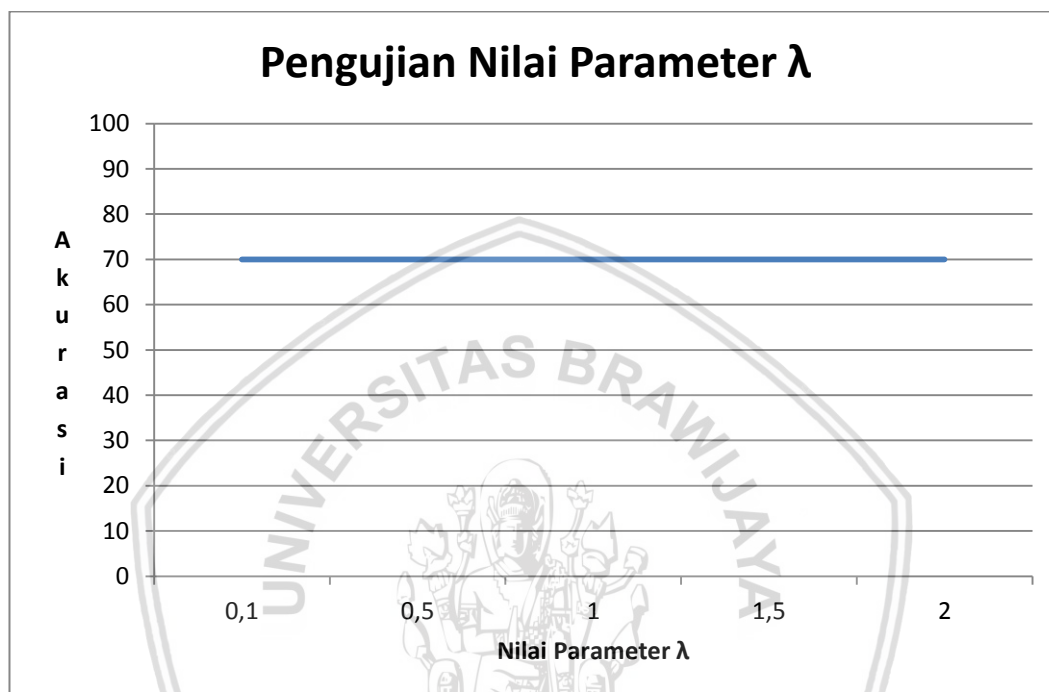
Tabel 5.6 Dokumen Relevan Pada Pengujian Terhadap Iterasi Maksimum

No.	Nomor Tweet	Prediksi Kelas	Cocok (Ya/Tidak)				
			Percobaan				
			1	2	3	4	5
1	1	Negatif	Ya	Ya	Ya	Ya	Ya
2	2	Negatif	Tidak	Ya	Tidak	Ya	Ya
3	3	Negatif	Ya	Ya	Ya	Ya	Ya
4	4	Negatif	Ya	Ya	Ya	Ya	Ya
5	5	Negatif	Ya	Ya	Ya	Ya	Ya
6	6	Negatif	Tidak	Tidak	Tidak	Tidak	Tidak
7	7	Negatif	Ya	Ya	Ya	Ya	Ya
8	8	Negatif	Ya	Tidak	Ya	Tidak	Ya
9	9	Negatif	Ya	Ya	Ya	Ya	Ya
10	10	Negatif	Tidak	Tidak	Tidak	Tidak	Tidak
11	11	Negatif	Tidak	Tidak	Tidak	Tidak	Tidak
12	12	Positif	Tidak	Tidak	Tidak	Tidak	Tidak
13	13	Positif	Ya	Ya	Ya	Ya	Ya
14	14	Positif	Ya	Ya	Ya	Ya	Ya
15	15	Negatif	Ya	Ya	Ya	Ya	Ya
16	16	Positif	Ya	Ya	Ya	Ya	Ya
17	17	Negatif	Tidak	Tidak	Tidak	Tidak	Tidak
18	18	Positif	Ya	Ya	Ya	Ya	Ya
19	19	Negatif	Ya	Ya	Ya	Ya	Ya
20	20	Negatif	Ya	Ya	Ya	Ya	Ya

### 5.3 Analisis Pengujian

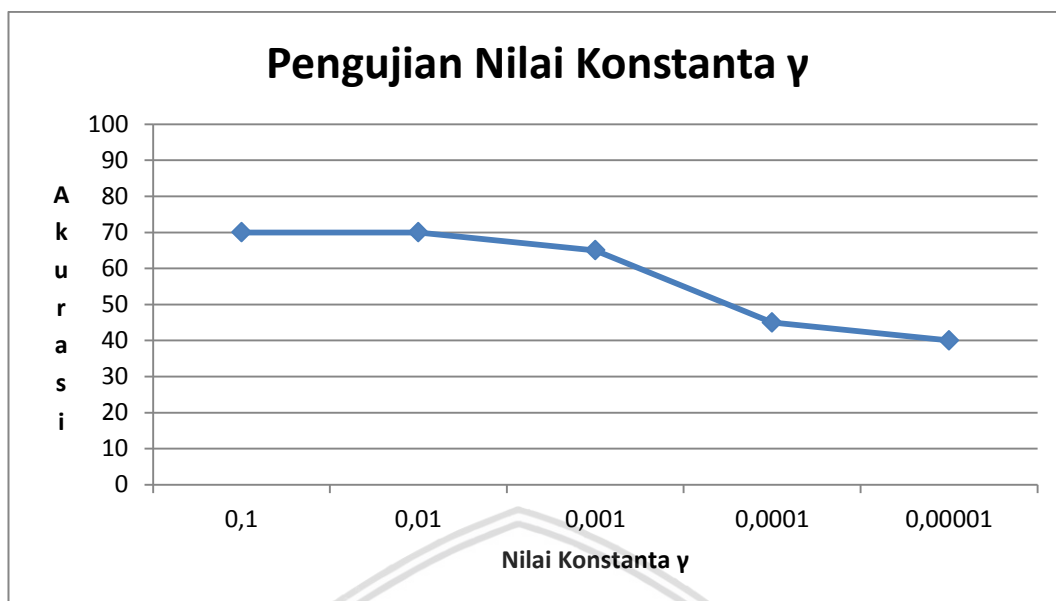
Pada pengujian nilai parameter  $\lambda$  dapat diambil kesimpulan bahwa nilai parameter  $\lambda$  tidak mempunyai pengaruh terhadap hasil akurasi. Hal ini dapat

dipengaruhi karena selisih nilai parameter satu dengan yang lain yang digunakan dalam pengujian sangat kecil sehingga perubahan akurasi tidak dapat ditunjukkan. Sehingga dapat disimpulkan bahwa nilai parameter yang digunakan dalam pengujian berada di dalam satu interval. Hal itu dapat dilihat pada nilai akurasi yang dijabarkan pada Tabel 5.1. Pada pengujian 1, pengujian 2, pengujian 3, pengujian 4, dan pengujian 5 akurasi yang diperoleh sebesar 70%. Nilai pada Tabel 5.1 direfleksikan pada Gambar 5.1.



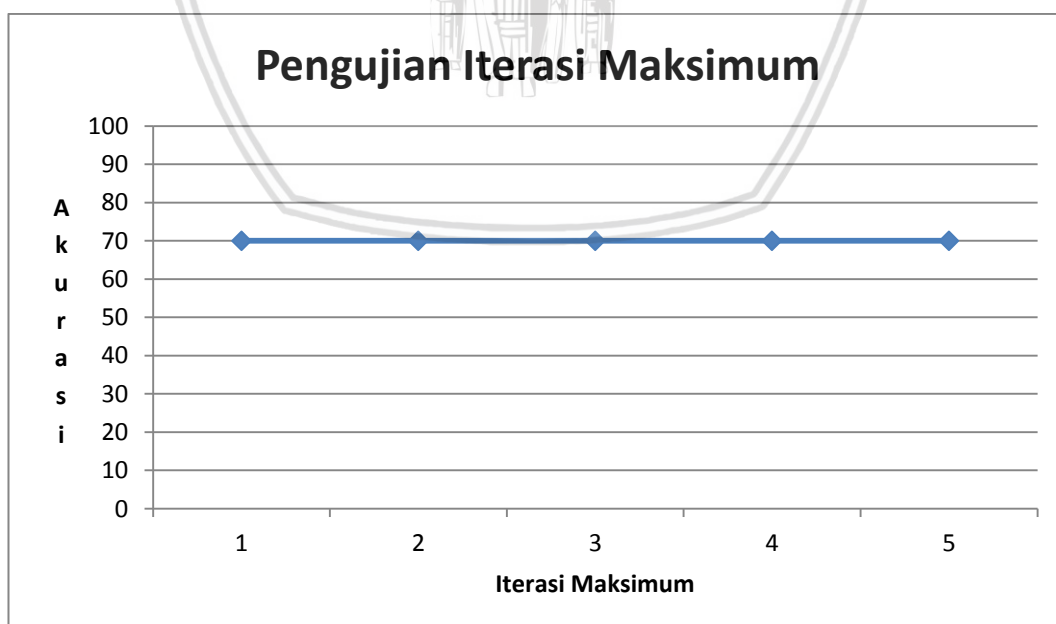
**Gambar 5.1 Pengujian Nilai Parameter  $\lambda$**

Pada pengujian nilai konstanta  $\gamma$  dapat diambil kesimpulan bahwa nilai konstanta  $\gamma$  mempunyai pengaruh terhadap hasil akurasi. Pada percobaan pertama dan kedua hasil akurasi pengujian tetap. Hal ini bisa disebabkan karena nilai parameter yang digunakan berada dalam satu interval. Kemudian pada percobaan selanjutnya hasil akurasi mengalami penurunan. Hal ini dikarenakan nilai parameter yang dipakai berada dalam interval yang berbeda dengan nilai parameter yang digunakan pada percobaan sebelumnya. Pengujian ini dapat disimpulkan bahwa semakin kecil nilai konstanta  $\gamma$ , maka akurasi akan menurun. Hal itu dapat dilihat pada nilai akurasi yang dijabarkan pada Tabel 5.3. Pengujian 1, pengujian 2, menghasilkan akurasi sebesar 70%. Pengujian 3 menghasilkan akurasi sebesar 65%. Pengujian 4 menghasilkan akurasi 45%. Pengujian 5 akurasi menghasilkan akurasi sebesar 40%. Nilai pada Tabel 5.3 direfleksikan pada Gambar 5.2.



**Gambar 5.2 Pengujian Nilai Konstanta  $y$**

Pada pengujian iterasi maksimum dapat diambil kesimpulan bahwa iterasi tidak mempunyai pengaruh terhadap hasil akurasi. Hal ini dapat dipengaruhi karena selisih nilai parameter satu dengan yang lain yang digunakan dalam pengujian sangat kecil sehingga perubahan akurasi tidak dapat ditunjukkan. Dan disimpulkan bahwa nilai parameter yang digunakan dalam pengujian berada di dalam satu interval. Hal itu dapat dilihat pada nilai akurasi yang dijabarkan pada Tabel 5.5. Pengujian 1, pengujian 2, pengujian 3, pengujian 4, dan pengujian 5 akurasi yang diperoleh sebesar 70%. Nilai pada Tabel 5.5 direfleksikan pada Gambar 5.3.



**Gambar 5.3 Pengujian Iterasi Maksimum**

## BAB 6 PENUTUP

Bab ini berisi kesimpulan dan saran dari penelitian yang telah dilakukan. Kesimpulan berisi tentang inti penelitian. Saran adalah masukan untuk pengembangan untuk penelitian lebih lanjut.

### 6.1 Kesimpulan

Dari penelitian yang telah dilakukan dapat diambil kesimpulan sebagai berikut.

1. Metode *Support Vector Machine* diimplementasikan untuk menganalisis sentimen terhadap konten radikal dokumen Twitter berupa *tweet* berbahasa Indonesia menjadi dua kelas (konten radikal positif dan konten radikal negatif) dengan menggunakan enam fitur *term* (kata benda positif, kata benda negatif, kata kerja positif, kata kerja negatif, kata sifat positif, dan kata sifat negatif).
2. Hasil akurasi terbaik berdasarkan pengujian yang dilakukan adalah sebesar 70% menggunakan nilai parameter  $\lambda$  sebesar 0,1, nilai konstanta  $\gamma$  sebesar 0,1, iterasi maksimum 5 dengan jumlah data sebanyak 80 dokumen (60 dokumen negatif dan 20 dokumen positif) untuk data latih dan 20 dokumen (15 dokumen negatif dan 5 dokumen positif) untuk data uji.

### 6.2 Saran

Dari penelitian yang telah dilakukan dapat diambil kesimpulan sebagai berikut.

1. Diperlukan penelitian lebih lanjut dengan menggunakan pendekatan semantik atau melihat dari urutan kata dan maknanya.
2. Diperlukan penelitian lebih lanjut mengenai kombinasi nilai parameter  $\lambda$ , konstanta  $\gamma$ , dan iterasi maksimum yang digunakan untuk mendapatkan akurasi yang lebih optimal.

## DAFTAR PUSTAKA

- Amin, F., 2013. *Sistem Temu Kembali Informasi dengan Pemeringkatan Metode Vector Space Model*. [Online] Tersedia di: <<http://download.portalgaruda.org/article.php?article=109855&val=544>> [Diakses 2 Februari 2016]
- Anggraeni, A., 2013. *Budaya Curhat di Jejaring Sosial*. [Online] Tersedia di: <[http://www.kompasiana.com/aryanggraeni/budaya-curhat-di-jejaring-sosial\\_552a2f1d6ea8347b32552d61](http://www.kompasiana.com/aryanggraeni/budaya-curhat-di-jejaring-sosial_552a2f1d6ea8347b32552d61)> [Diakses 2 Februari 2016]
- Anggraeni, I., 2013. *Internet Kini Menjadi Kebutuhan Primer*. [Online] Tersedia di: <[http://www.kompasiana.com/intananggraeni/internet-kini-menjadi-kebutuhan-primer\\_5528a38cf17e61416f8b4567](http://www.kompasiana.com/intananggraeni/internet-kini-menjadi-kebutuhan-primer_5528a38cf17e61416f8b4567)> [Diakses 2 Februari 2016]
- Anonymous. 2012. *Soal Radikalisme*. [Online] Tersedia di: <<http://rubik.okezone.com/read/8170/soal-radikalisme>> [Diakses 2 Februari 2016].
- Asih, R., 2012. *Indonesia Pengguna Twitter Terbesar Kelima Dunia*. [Online] Tersedia di: <<http://tekno.tempo.co/read/news/2012/02/02/072381323/indonesia-pengguna-twitter-terbesar-kelima-dunia>> [Diakses 2 Februari 2016]
- Husaini, A., 2005. *Radikalisme dan Terorisme*. [Online] Tersedia di: <<http://www.hidayatullah.com/kolom/catatan-akhir-pekan/read/2005/11/27/3540/radikalisme-dan-terorisme.html>> [Diakses 2 Februari 2016]
- Iradat, Damar., 2015. *Ini Empat Kriteria Situs Radikal Menurut BNPT*. [Online] Tersedia di: <<http://news.metrotvnews.com/read/2015/04/05/381416/ini-empat-kriteria-situs-radikal-menurut-bnpt>> [Diakses 30 Juni 2016].
- Khan, A. Z. H., 2015. *Sentiment Analysis Using Support Vector Machine*. [Online] Tersedia di: <[http://www.ijarcsse.com/docs/papers/Special\\_Issue/ITSD2015/25.pdf](http://www.ijarcsse.com/docs/papers/Special_Issue/ITSD2015/25.pdf)> [Diakses 2 Februari 2016].
- Kurniawati, M., 2014. *Klasifikasi Dokumen E-Complaint Kampus Menggunakan Directed Acyclic Graph Multi-Class SVM*. Skripsi Mahasiswa PTIIK Universitas Brawijaya.
- Kustiani, R., 2011. *Survei Internet Termasuk Kebutuhan Pokok*. [Online] Tersedia di: <<http://tekno.tempo.co/read/news/2011/09/26/072358326/survei-internet-termasuk-kebutuhan-pokok>> [Diakses 2 Februari 2016].
- Liu, B., 2010. *Sentiment Analysis and Subjectivity*. [Online] Tersedia di: <<https://www.cs.uic.edu/~liub/FBS/NLP-handbook-sentiment-analysis.pdf>> [Diakses 2 Februari 2016].



- Liu, B., 2012. *Sentiment Analysis and Opinion Mining*. [Online] Tersedia di: <<https://www.cs.uic.edu/~liub/FBS/SentimentAnalysis-and-OpinionMining.pdf>> [Diakses 2 Februari 2016].
- Merdekawati, N. N. D., 2014. *Analisis Sentimen Pencitraan Elite Politik Berdasarkan Opini Melalui Media Sosial Twitter Menggunakan Metode Additive Kernel SVM*. Skripsi Mahasiswa PTIK Universitas Brawijaya.
- Novantirani, A., 2015. *Analisis Sentimen pada Twitter untuk Mengenai Penggunaan Transportasi Umum Darat Dalam Kota dengan Metode Support Vector Machine*. [Online] Tersedia di: <[https://repository.telkomuniversitas.ac.id/pustaka/files/100653/jurnal\\_eproc/analisis-sentimen-pada-twitter-mengenai-penggunaan-transportasi-umum-darat-dalam-kota-dengan-metode-support-vector-machine.pdf](https://repository.telkomuniversitas.ac.id/pustaka/files/100653/jurnal_eproc/analisis-sentimen-pada-twitter-mengenai-penggunaan-transportasi-umum-darat-dalam-kota-dengan-metode-support-vector-machine.pdf)> [Diakses 2 Februari 2016].
- O'Reilly, T. & Milstein, S., 2011. *The Twitter Book*. 2<sup>nd</sup> ed. [Online] Tersedia di: <[http://leotardi.ddns.info/download/The\\_Twitter\\_Book.pdf](http://leotardi.ddns.info/download/The_Twitter_Book.pdf)> [Diakses 2 Februari 2016].
- Pang, B. & Lee, L., 2008. *Opinion Mining and Sentiment Analysis*. [Online] Tersedia di: <<http://www.cs.cornell.edu/home/llee/omsa/omsa.pdf>> [Diakses 2 Februari 2016].
- Patil, G. et al., 2014. *Sentiment Analysis Using Support Vector Machine*. [Online] Tersedia di: <[http://www.ijircce.com/upload/2014/january/16K\\_Sentiment.pdf](http://www.ijircce.com/upload/2014/january/16K_Sentiment.pdf)> [Diakses 2 Februari 2016].
- Putranti, N. D., 2014. *Analisis Sentimen Twitter untuk Teks Berbahasa Indonesia dengan Maximum entropy dan Support Vector Machine*. [Online] Tersedia di: <<http://journal.ugm.ac.id/ijccs/article/viewFile/3499/3025>> [Diakses 2 Februari 2016].
- Souza, C., 2010. *Kernel Functions for Machine Learning Applications*. [Online] Tersedia di: <<http://crsouza.com/2010/03/kernel-functions-for-machine-learning-applications>> [Diakses 14 Februari 2016]
- Tala, F. Z., 2003. *A Study of Stemming Effects on Information Retrieval in Bahasa Indonesia*. [Online] Tersedia di: <<https://www.illc.uva.nl/Research/Publications/Reports/MoL-2003-02.text.pdf>> [Diakses 2 Februari 2016].
- Vijayakumar, S. & Wu, S., 1999. *Sequential Support Vector Machine*. [Online] Tersedia di: <<http://homepages.inf.ed.ac.uk/svijayak/publications/vijayakumar-SOCO1999.pdf>> [Diakses 14 Februari 2016].